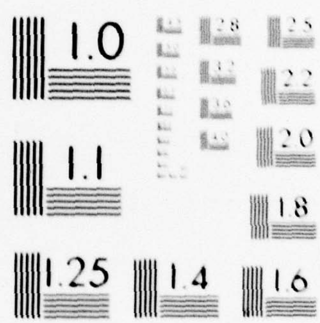
AD-A074 758 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 9/2
SESAME-A SYSTEM OF EQUATIONS FOR THE SIMULATION OF AIRCRAFT IN --ETC(U)
JAN 79 B N TOMLINSON
UNCLASSIFIED RAE-TR-79008 DRIC-BR-68659 NL

1 OF 2
AD
A074758



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

UNLIMITED

TR 79008
BR68659

(18) DRIC
(19) BR-68659

1
B.S.

ROYAL AIRCRAFT ESTABLISHMENT

LEVEL II

*

(9) Technical ~~Report~~, 79008

(11) January 1979 (14) RAE-TR-79008

(6)
SESAME—A SYSTEM OF EQUATIONS FOR THE SIMULATION OF AIRCRAFT IN A MODULAR ENVIRONMENT.

by (12) 152

(10) B.N. Tomlinson

*

Procurement Executive, Ministry of Defence
Farnborough, Hants

UNLIMITED 310 450

DISCLAIMER NOTICE

UDC 5.001.57 : 5.001.58 : 531.31 : 519.688

ROYAL AIRCRAFT ESTABLISHMENT

Technical Report 79008

Received for printing 12 January 1979

SESAME - A SYSTEM OF EQUATIONS FOR THE SIMULATION OF AIRCRAFT IN A MODULAR ENVIRONMENT

by

B. N. Tomlinson

SUMMARY

A system of equations has been developed for the simulation of an aircraft's motion in real time using a digital computer. Those parts of the mathematical model common to all aircraft have been created as a set of Fortran subroutines, leaving the user to create only a small group of routines specifically to describe his aircraft. The equations employed are defined and the computer implementation described in detail. The Report can be used as a handbook and 'user guide' but as the routines described are not specific to real-time simulation they could be used as a basis for a general mathematical model of an aircraft for use on any computer which supports Fortran.

Departmental Reference: FS 91

LIST OF CONTENTS

LIST OF CONTENTS (concluded)

1 INTRODUCTION

This Report describes a system of equations for the simulation of an aircraft's 'rigid-body' motion in real time using a digital computer. Known as SESAME - a System of Equations for the Simulation of Aircraft in a Modular Environment - it may be described as opening the door to flexible use of the simulator. Those parts of the mathematical model which are common to all aircraft, such as the equations of motion and axis transformations, have been created as a set of standard modules (written in Fortran), leaving the user to create only a small group of routines specifically to describe his aircraft. The two sets of modules are then linked together to produce a complete model program.

The simulator for which SESAME has been created changed from pure analogue to hybrid computing[1] in 1974. The computer system consists of a Xerox Sigma 8 digital computer linked to an Applied Dynamics AD4 analogue computer. This Report is concerned mainly with the digital computer aspects. In the early days of using this new computing facility, the aircraft's mathematical model, computed entirely digitally, was coded using SL1, a simulation language for solving sets of ordinary differential equations. With increasing experience and increasingly complex simulations, the single large program imposed by the SL1 language was found to be more and more inconvenient and the decision was taken to restructure the program into small, self-contained modules.

Such restructuring became feasible largely because a newly-created software package, for on-line parameter variation and inspection, provided the means to access and change, on-line, variables contained in subroutines.

Although the basic modules are themselves coded in Fortran, SL1 itself has been retained to provide the overall framework in which the modules sit. This is because SL1 provides facilities for numerical integration and for synchronisation with real-time.

The structure of the Report is as follows. After a discussion, in section 2, of the objectives to be achieved by the modular nature of SESAME, section 3 gives a brief theoretical background to the system equations, including such topics as choice of axes and definition of the equations of motion. This is followed in section 4 by an outline of how the equations to be solved are distributed among the specific routines. This is supplemented (in Appendix A) by a detailed definition of each routine and, as the ultimate specification for reference purposes,
8 by a listing of the routines (Appendix B). Section 5 then describes the overall program structure in which the routines are embedded. Section 6 discusses how

communication among routines is achieved using the Fortran named COMMON facility, and how the user should adopt the same technique for his routines. Hybrid input/ output (analogue to digital and digital to analogue conversion, and single bit discretes) is also described in section 6 in terms of what the user receives and must provide, while section 7 is a general "User's Guide", summarising what the system expects from the user, defining the set of routines and the data the user must provide, and outlining the choices the user may make regarding the atmosphere, turbulence and other features.

The Report has been written with three principal aims: to provide primary simulator users with a description of the standard computer model and an outline of how to build it into a full simulation; to provide simulator customers with an idea of how the model is formulated and what must be done to represent an individual aircraft, so that they can, if necessary, contribute to the creation of a complete simulation model; and to provide a general description of a computer model for others interested in doing something similar.

2 OBJECTIVES

Specific objectives to be achieved by the modular system of equations concern size, testing, transferability, communication and execution time, and are intended to overcome detailed weaknesses of the simulation language SL1. However, as explained later in section 5.1, SL1 is retained as the appropriate language for the overall program, containing the individual modules.

2.1 Size

A complete model program is large, often exceeding 20000 words of computer storage. If coded entirely in SL1 source language this means many lines of code. The process of translating the SL1 source code into Fortran and then compiling is slow and tedious and must be repeated in its entirety whenever a change is made at source level. The present scheme drastically cuts the size of the SL1 source program, which in itself will speed up the translation/compilation process. In addition, changes at SL1 level will be rare, since the detail of a simulation is relegated to a lower level of routine. The existence of subroutines also provides the potential ability to create overlays, in order to reduce main store occupancy.

2.2 Creation and testing

Creation and checking of modules will be easier than handling a large SL1 program since routines, such as for the generation of the aircraft's aerodynamic forces, can be defined, created and tested in isolation before being included in

the main program. Work could proceed in parallel on several routines, and be done by external users of the simulator, *ie* people who are not specialists in simulation as such.

Modifications will also be easier, since they should involve only revision of a single subroutine, its compilation and then creation of a new load module without having to repeat the SL1 translation.

Using Fortran rather than SL1 as the primary medium does not mean that all the useful SL1 facilities, such as function generation by table look-up, must be discarded. These can still be used at the Fortran level, since each one involves a Fortran-callable subroutine or function. However, the calling sequence will, in general, be more complicated in Fortran than in SL1. SL1 operators involving integration are not usable in subroutines without considerable effort.

2.3 Transferability

Having a modular structure means that simulation of a new aircraft requires only the creation of the routines specific to that aircraft. The standard parts can be picked up from libraries and special facilities, such as simulation of guidance beams, can readily be moved from simulation to simulation in subroutine form. With SL1 the only way of transferring across an existing piece of code is by explicit inclusion of the source lines and punching of the necessary cards.

None of the modules includes any compromising feature which limits its employment to real-time simulation. Hence the routines described could also form the basis for a conventional digital model of aircraft dynamics for use in non-real-time studies.

2.4 Communication

A defect of the SL1 language is the way most of the significant variables are lumped into one large block of labelled COMMON, the name of which is not fixed and the contents of which can change if the SL1 source code is varied.

In this new system communication of variables among all the modules is achieved by using Fortran labelled COMMON, but under *user* control. This retains the advantage of using COMMON storage, that variables are conveniently accessible by any subroutine without using arguments, thus saving execution time.

2.5 Execution time

The aircraft's motion is computed by repeatedly integrating the differential equations. If a multi-pass integration technique is used, such as Runge-Kutta fourth order, there are some sections of the equations which are not part

of the dynamic loop and so need only be executed once per time step. Examples are calculation of ILS guidance and TV position signals. By having these elements as subroutines, it is easy to arrange that they are executed only as often as necessary and so save time for more important functions.

To date, aircraft model programs have been solved with only one basic loop or frame. SL1 does provide a means to execute one part at a different iteration rate from another. Should circumstances arise where more than one frame time is necessary, recasting of the model program (at SL1 skeleton level) will be aided by the simplicity of merely moving subroutine calls from one 'derivative' section to another, rather than blocks of code which would be necessary in a program coded completely in SL1. Communication between 'derivative' sections is also no problem when variables are in COMMON areas controlled by the user rather than by the SL1 translator.

2.6 Data logging

Access to variables for data logging purposes is easy and convenient. A self-contained data logging routine can be written knowing that all variables may be accessed without special action and without run-time inefficiencies.

3 SYSTEM EQUATIONS

3.1 Choice of axes

The first requirement is to choose sets of axes in which to solve the fundamental equations of motion. In classic texts[2-4] a set of axes fixed in the aircraft is generally chosen but, being a rotating frame of reference, such body axes result in the translational accelerations including angular velocity terms, as in

$$X - mg \sin \theta \; = \; m(\dot{u} + qw - rv) \; .$$

For efficient computer solution of the equations of motion, it is desirable to uncouple translational motion from rotational motion, a point which was made many years ago by Howe[5] and reiterated recently[6]. This may be achieved by suitable choice of axes. Then, since rotational motion of aircraft is intrinsically more rapid than translational motion, solution of the rotational equations may be performed more frequently than the translational equations without an excessive computing load, and while maintaining overall accuracy.

Body-fixed axes still provide a natural frame for the solution of rotational equations of motion, with the advantage of constant moments of inertia. Choice 008

of axes therefore reduces to selecting an appropriate frame for the translational equations of motion.

So-called 'flight path axes' could be a suitable choice, with the origin at the aircraft's centre of gravity and the x axis aligned with the aircraft's velocity vector with respect to the ground. However, in the presence of winds and turbulence this introduces complications, so that, following Ref 7, the choice falls on earth-based axes.

3.2 Definition of axes

All axes systems used are orthogonal, right-handed triads.

3.2.1 Earth axes

Earth axes are an inertial frame assuming a flat, non-rotating earth. (See McFarland[7] for the case of a spherical, rotating earth.)

The origin is at a datum point on the visual model* in use, typically at the runway threshold and on the centreline. The x-axis points northward (suffix N), the y-axis points eastward (suffix E), the xy plane being parallel to the earth's surface, and the z-axis (suffix D) points down to the centre of the earth.

3.2.2 Geometric body axes

Geometric body axes have their origin at the aircraft's centre of gravity and are located with respect to the aircraft by some geometric feature such as the longitudinal fuselage datum line. Once defined they are fixed in the aircraft. The x-axis points forward, the y-axis to starboard and the z-axis 'down'.

3.3 Aircraft attitude

Aircraft angular orientation with respect to the earth is defined[8] by a conventional trio of Euler angles ψ, θ, ϕ . The heading angle ψ is measured from north and lies in the range 0, 360°; the elevation angle or pitch attitude θ is measured from the horizontal plane and lies in the range -90, $+90^{\circ}$; the roll (or bank) angle is measured from the horizontal plane and lies in the range -180, $+180^{\circ}$.

Rates of change of these angles may be related to the components of angular velocity of the aircraft by

08

* Outside world cues are provided by a closed-circuit TV system viewing a physical model of an appropriate terrain.

$$\left.\begin{aligned} \dot{\theta} &= q \cos\phi - r \sin\phi \\ \dot{\phi} &= p + (q \sin\phi + r \cos\phi) \tan\theta = p + \dot{\psi} \sin\theta \\ \dot{\psi} &= (q \sin\phi + r \cos\phi) \sec\theta \end{aligned}\right\} \quad (1)$$

and the angles themselves obtained by integration. However, a singularity occurs at $\theta = 90^{\circ}$. If all-attitude manoeuvring is desired, an alternative formulation of equations is necessary. No provision has yet been made for multiple rolls or turns.

3.4 Transformation from earth to body axes

Transformation of a set of variables from earth axes to body axes (or vice versa) is most conveniently achieved through the direction cosine matrix[4,9]. For example, the components of airspeed in body axes (u_B, v_B, w_B) are related to the components in earth axes (V_N, V_E, V_D) by

$$\left.\begin{aligned} u_B &= \ell_1 V_N + \ell_2 V_E + \ell_3 V_D \\ v_B &= m_1 V_N + m_2 V_E + m_3 V_D \\ w_B &= n_1 V_N + n_2 V_E + n_3 V_D \end{aligned}\right\} \quad (2)$$

where ℓ_1 etc are the direction cosines, given by

$$\left.\begin{aligned} \ell_1 &= S_{11} = \cos\theta \cos\psi \\ \ell_2 &= S_{12} = \cos\theta \sin\psi \\ \ell_3 &= S_{13} = -\sin\theta \\ m_1 &= S_{21} = \sin\phi \sin\theta \cos\psi - \cos\phi \sin\psi \\ m_2 &= S_{22} = \sin\phi \sin\theta \sin\psi + \cos\phi \cos\psi \\ m_3 &= S_{23} = \sin\phi \cos\theta \\ n_1 &= S_{31} = \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi \\ n_2 &= S_{32} = \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi \\ n_3 &= S_{33} = \cos\phi \cos\theta \ . \end{aligned}\right\} \quad (3)$$

In matrix form, equations (2) may be written

$$\left\{u_B\right\} = S\left\{V_N\right\} \tag{4}$$

where $\left\{u_B\right\}$, $\left\{V_N\right\}$ are column matrices (after Hopkin[4]) and S , the transformation matrix, is orthogonal, its transpose S^T being the same as its inverse.

The inverse relationship, for earth axes variables in terms of body axis variables, is then given by

$$\left\{V_N\right\} = S^T\left\{u_B\right\} . \tag{5}$$

Thus, for example, the total forces (suffix T) applied in body axes (suffix X, Y, Z) may be transformed to earth axes (suffix N, E, D) by

$$\left\{F_{TN}\right\} = S^T\left\{F_{TX}\right\} \tag{6}$$

which, expanded, is

$$\left.\begin{aligned} F_{TN} &= S_{11}F_{TX} + S_{21}F_{TY} + S_{31}F_{TZ} \\ F_{TE} &= S_{12}F_{TX} + S_{22}F_{TY} + S_{32}F_{TZ} \\ F_{TD} &= S_{13}F_{TX} + S_{23}F_{TY} + S_{33}F_{TZ} \quad . \end{aligned}\right\} \tag{7}$$

Other vector components are related in the same way.

3.5 Equations of motion - translation

The components of acceleration with respect to the earth are obtained from

$$\left.\begin{aligned} \dot{V}_{KN} &= F_{TN}/m \\ \dot{V}_{KE} &= F_{TE}/m \\ \dot{V}_{KD} &= F_{TD}/m + g \end{aligned}\right\} \tag{8}$$

where the acceleration due to gravity, g , is assumed constant, and the forces F_{TN} etc in earth axes are obtained from the total forces in body axes F_{TX}, F_{TY}, F_{TZ} by the equations (7).

Integration of equations (8) yields the velocity components (V_{KN}, V_{KE}, V_{KD}) of the vehicle relative to the earth. Relative to the air mass, the velocity
)8 components of the aircraft are

$$\left.\begin{aligned} V_N &= V_{KN} - V_{WN} \\ V_E &= V_{KE} - V_{WE} \\ V_D &= V_{KD} - V_{WD} \end{aligned}\right\} \tag{9}$$

where V_{WN}, V_{WE}, V_{WD} are the components of the wind velocity relative to the ground.

The body-axes components of the velocity vector relative to the air may then be derived using equations (2) and (9), from which the velocity (or True Airspeed, TAS) is

$$V_T = \left(u_B^2 + v_B^2 + w_B^2\right)^{\frac{1}{2}} \tag{10}$$

and equivalent airspeed is derived using the atmospheric density ratio

$$V = V_T \sigma^{\frac{1}{2}} \quad . \tag{11}$$

Angles of attack and sideslip are computed from[10]

$$\alpha = \tan^{-1} w_B/u_B \tag{12}$$

(α being in the range -180, $+180^{\circ}$) and

$$\beta = \tan^{-1} v_B \Big/ \left(u_B^2 + v_B^2\right)^{\frac{1}{2}} \tag{13}$$

(β being in the range -90, $+90^{\circ}$ and taking the sign of v_B).

Derivatives of these angles with respect to time, needed to calculate the aerodynamic forces and moments, are created by simple difference equations. Flight path angles defining climb (γ) and track (χ) are derived from the velocity vector components relative to the ground, having first obtained the ground speed (V_K) as the horizontal component

$$V_K = \left(V_{KN}^2 + V_{KE}^2\right)^{\frac{1}{2}} \tag{14}$$

$$\gamma = \tan^{-1}\left(-V_{KD}/V_K\right) \tag{15}$$

$$\chi = \tan^{-1}\left(V_{KE}/V_{KN}\right) . \tag{16}$$

Positional coordinates x, y, h of the aircraft's centre of gravity are found by integrating the velocities

$$\left.\begin{aligned} \dot{x} &= V_{KN} \\ \dot{y} &= V_{KE} \\ \dot{h} &= -V_{KD} \ . \end{aligned}\right\} \quad (17)$$

A block diagram of the translation equations is illustrated in Fig 1.

3.6 Equations of motion - rotation

It is assumed that the x and z body axes lie in a plane of mass symmetry so that products of inertia I_{yz} and I_{xy} are zero. Then the classical equations (Hopkin[4], section 10.1) may be manipulated to give

$$\left.\begin{aligned} \dot{p} &= \left\{ L + \frac{I_{zx}}{I_z} N + \left[\frac{I_{zx}}{I_z}(I_z + I_x - I_y)\right] pq \right. \\ &\qquad \left. + \left[(I_y - I_z) - \frac{I_{zx}^2}{I_z}\right] qr \right\} \Bigg/ \left\{ I_x \left(1 - \frac{I_{zx}^2}{I_x I_z}\right) \right\} \\ \dot{q} &= \left\{ M + I_{zx}(r^2 - p^2) + (I_z - I_x) rp \right\} \Big/ I_y \\ \dot{r} &= \left\{ N + \frac{I_{zx}}{I_x} L + \left[(I_x - I_y) + \frac{I_{zx}^2}{I_x}\right] pq \right. \\ &\qquad \left. + \left[\frac{I_{zx}}{I_x}(-I_x + I_y - I_z)\right] qr \right\} \Bigg/ \left\{ I_z \left(1 - \frac{I_{zx}^2}{I_x I_z}\right) \right\} \end{aligned}\right\} \quad (18)$$

or

$$\left.\begin{aligned} \dot{p} &= CI_1 L + CI_2 N + (CI_3 p + CI_4 r) q \\ \dot{q} &= CI_5 M + CI_6 (r^2 - p^2) + CI_7 rp \\ \dot{r} &= CI_8 N + CI_2 L + (CI_9 p + CI_{10} r) q \end{aligned}\right\} \quad (19)$$

where CI_1 etc are constants evaluated during initialisation.

$$\left.\begin{aligned}
CI_1 &= I_z \big/ \left(I_x I_z - I_{zx}^2\right) \\
CI_2 &= I_{zx} \big/ \left(I_x I_z - I_{zx}^2\right) \\
CI_3 &= I_{zx}\left(I_x - I_y + I_z\right) \big/ \left(I_x I_z - I_{zx}^2\right) \\
CI_4 &= \left(I_z(I_y - I_z) - I_{zx}^2\right) \big/ \left(I_x I_z - I_{zx}^2\right) \\
CI_5 &= 1/I_y \\
CI_6 &= I_{zx} \big/ I_y \\
CI_7 &= \left(I_z - I_x\right) \big/ I_y \\
CI_8 &= I_x \big/ \left(I_x I_z - I_{zx}^2\right) \\
CI_9 &= \left(I_x(I_x - I_y) + I_{zx}^2\right) \big/ \left(I_x I_z - I_{zx}^2\right) \\
CI_{10} &= -I_{zx}\left(I_x - I_y + I_z\right) \big/ \left(I_x I_z - I_{zx}^2\right) \quad .
\end{aligned}\right\} \qquad (20)$$

No terms are included in these equations (18) and (19) to allow for engine gyroscopic effects, but they could easily be added by the user when he provides the total moments L, M, N . It would also be straightforward to extend the equations to allow for a non-symmetric mass distribution.

Solution of the equations of rotational motion therefore consists in calculating the angular acceleration components from equations (19) given the total moments L, M, N ; integrating to obtain angular velocity components; transforming, using equations (1), to attitude rates; integrating again to attitude angles and calculating the direction cosines according to equations (3). This total process is illustrated in Fig 2.

3.7 Centre of gravity location

An aircraft's centre of gravity location is usually defined, in a fore-and-aft sense, in terms of a reference length such as the mean chord $\bar{c}$. Thus if the centre of gravity is quoted as being at $0.1\bar{c}$, then it is 10% of the reference chord *aft* of the origin. This convention is in a direction contrary to the normal 008
positive sense of x, but will be retained here. The vertical location of the

centre of gravity, however, is defined as being measured in the positive z direction (*ie* downwards) from some origin.

Aerodynamic moment data may often be quoted relative to some reference centre of gravity position, so that for an actual operational centre of gravity position, some correction is necessary. This is provided for as follows. If the moment reference position of the centre of gravity is $x_{cg\,ref}$, $z_{cg\,ref}$ and the actual centre of gravity position is x_{cg}, z_{cg}, all defined as fractions of some reference length c_{ref} then the differences are calculated as

$$\left.\begin{aligned} \Delta x_{cg} &= x_{cg} - x_{cg\,ref} \\ \Delta z_{cg} &= z_{cg} - z_{cg\,ref} \ . \end{aligned}\right\} \qquad (21)$$

Then differences are then available to correct aerodynamic moments to the current centre of gravity position.

3.8 Pilot position and positional rates

Given the coordinates x, y, h of the aircraft's centre of gravity, the pilot's position, needed to drive outside world displays, is obtained from

$$\left.\begin{aligned} x_{TV} &= x + (x_{pcg} \cos\theta + z_{pcg} \sin\theta) \cos\psi \\ y_{TV} &= y + (x_{pcg} \cos\theta + z_{pcg} \sin\theta) \sin\psi \\ h_{TV} &= h + (x_{pcg} \sin\theta - z_{pcg} \cos\theta) \end{aligned}\right\} \qquad (22)$$

where

$$\left.\begin{aligned} x_{pcg} &= x_p + \Delta x_{cg} c_{ref} \\ z_{pcg} &= z_p - \Delta z_{cg} c_{ref} \end{aligned}\right\} \qquad (23)$$

and x_p, z_p are the ooordinates of the pilot's eye point, assumed to be on the aircraft centre line, and defined relative to the reference centre of gravity position.

If we put

$$\left.\begin{aligned} R_{xp} &= x_{pcg} \cos\theta + z_{pcg} \sin\theta \\ R_{hp} &= x_{pcg} \sin\theta - z_{pcg} \cos\theta \end{aligned}\right\} \qquad (24)$$

)08

then

$$\left.\begin{aligned} \dot{x}_{TV} &= \dot{x} - R_{xp}\dot{\psi}\sin\psi - R_{hp}\dot{\theta}\cos\psi \\ \dot{y}_{TV} &= \dot{y} + R_{xp}\dot{\psi}\cos\psi - R_{hp}\dot{\theta}\sin\psi \\ \dot{h}_{TV} &= \dot{h} + R_{xp}\dot{\theta} . \end{aligned}\right\} \quad (25)$$

3.9 Accelerations in body axes

Linear accelerations at the aircraft's centre of gravity, at the pilot's head and elsewhere (*eg* for accelerometers) are needed.

Referred to body axes, the components (in units of g) of the acceleration of the aircraft's centre of gravity are

$$\left.\begin{aligned} a_{xcg} &= F_{TX}/W \\ a_{ycg} &= F_{TY}/W \\ a_{zcg} &= F_{TZ}/W . \end{aligned}\right\} \quad (26)$$

These are referred to as the 'specific' accelerations and are the acceleration components that would be measured by a set of orthogonal, body-fixed, accelerometers aligned with the body axes and located at the centre of gravity of the aircraft. 'Absolute' accelerations, including the gravity components, are then

$$\begin{aligned} a_{xacg} &= a_{xcg} - \sin\theta &&= a_{xcg} + S_{13} \\ a_{yacg} &= a_{ycg} + \sin\phi\cos\theta &&= a_{ycg} + S_{23} \\ a_{zacg} &= a_{zcg} + \cos\phi\cos\theta &&= a_{zcg} + S_{33} . \end{aligned}$$

It should be remembered that a_{xacg} is not equal to $\dot{u}_b$ because the body axis system is rotating, as explained in any text, such as Etkin[3], Chapter 4.

At an arbitrary location $L(x_L, y_L, z_L)$, the components of specific acceleration are

$$\left.\begin{aligned} a_{x_L} &= a_{xcg} - \left[x_L(q^2 + r^2) - y_L(pq - \dot{r}) - z_L(pr + \dot{q})\right]\Big/g \\ a_{y_L} &= a_{ycg} + \left[x_L(pq + \dot{r}) - y_L(p^2 + r^2) + z_L(qr - \dot{p})\right]\Big/g \\ a_{z_L} &= a_{zcg} + \left[x_L(pr - \dot{q}) + y_L(qr + \dot{p}) - z_L(p^2 + q^2)\right]\Big/g . \end{aligned}\right\} \quad (27)$$

The indication of an accelerometer or a 'g' meter may then be derived from equations (27) given the location of the device. Similarly the linear acceleration components at the pilot's station (x_p, z_p) may be derived for use in driving motion systems. Further details of the accelerations computed are given in the description of the SACCBOD routine in Appendix A.

3.10 Wind, wind shear and turbulence

Wind

A datum mean wind is defined in speed and direction by V_{WKTO} and ψ_W, from which the components in earth axes are obtained as

$$\left.\begin{aligned} V_{WNLO} &= -V_{WKTO} \cos \psi_W \\ V_{WELO} &= -V_{WKTO} \sin \psi_W \end{aligned}\right\} \qquad (28)$$

bearing in mind the convention that when ψ_W is zero, the wind is *from* the north. The vertical component of mean wind, V_{WDLO}, is normally assumed to be zero.

Wind shear

Wind shear is obtained as a multiplying factor f according to altitude, and the wind components at height are then

$$\left.\begin{aligned} V_{WNL} &= fV_{WNLO} \\ V_{WEL} &= fV_{WELO} \ . \end{aligned}\right\} \qquad (29)$$

This shear is effective only in magnitude, not in direction. Three choices are available: no shear (f = 1.0), logarithmic profile and linear profile. The initialisation process allows for actual wind at the initial height and sets up the aircraft's track to give a desired initial heading.

Turbulence

Turbulence in three orthogonal directions can be added to the components of the mean wind to give total wind components. Gust velocity components u_G, v_G, w_G are calculated by a new technique[12], based on the Statistical Discrete Gust theory of J.G. Jones[13], which allows the intermittent character of the generated turbulence to be controlled. Scaled components of turbulence u_T, v_T, w_T are obtained by

$$\left.\begin{aligned} u_T &= u_{sig} u_G \\ v_T &= v_{sig} v_G \\ w_T &= w_{sig} w_G \end{aligned}\right\} \qquad (30)$$

where u_{sig}, v_{sig}, w_{sig} are desired root-mean-square intensities. These turbulence components are considered to be along the wind, across the wind and vertical, so that the total fluctuating wind components, in earth axes, are

$$\left.\begin{aligned} V_{WN} &= V_{WNL} - (u_T \cos \psi_W - v_T \sin \psi_W) \\ V_{WE} &= V_{WEL} - (v_T \cos \psi_W + u_T \sin \psi_W) \\ V_{WD} &= V_{WDL} - w_T \quad . \end{aligned}\right\} \qquad (31)$$

3.11 Properties of the atmosphere

The properties of the ICAO International Standard Atmosphere[11] are calculated up to a maximum height of 65616 ft by the routine SATMOS (see description of SVELOC2 in Appendix A). Given altitude, it returns the density ratio, speed of sound ratio, temperature ratio and pressure ratio for a standard day. A routine ATMOS is also available for hot days but this is not integrated into SESAME. Strictly the altitude input should be 'geopotential' but below 65616 ft it is adequate to treat geometric and geopotential altitude as interchangeable.

The user may optionally choose to use constant atmospheric properties (as in many classical studies) or else allow standard variation with altitude. A software flag KISA controls this option.

4 TECHNICAL IMPLEMENTATION

Section 3 has defined a set of equations which compose the standard processes of any aircraft simulation. These equations are distributed among a series of modules for solution by digital computer. This section briefly describes the routines and their functions. Full details of each routine are given in Appendix A and listings are given in Appendix B. In general, system routines have names beginning with S . The routines fall into four categories:

Initialisation	- SINIT, SYSCOM
Rotational motion	- SDCOS, SEULER, SACCROT
Translational motion	- SVELOC1, SALFBET, SACCLIN, SPATH, SVELOC2
Utility functions	- STV, SILS, SWIND, SACCBOD, SATMOS, SCOUNT .

All the subroutines described in this Report are coded in Fortran. The particular dialect used is Xerox Extended Fortran IV, designed for the Xerox Sigma range of computers. It was not intended to use only those features of the language embodied in Standard Fortran IV and a few non-standard features have in fact been used. These include

variable names of up to 8 characters,
initialisation of COMMON variables within subroutines,
use of the NAMELIST feature (in SYSCOM only).

Conversion of the subroutines to run on another computer should, however, pose few problems.

4.1 Initialisation

Two routines (SINIT, SYSCOM) are specifically concerned with performing certain start-up, or initialisation, functions. Other initialisation functions may be performed internally in the other routines.

SINIT calculates various aircraft related constants, sets up initial height, derives initial values of atmospheric properties and *initialises speeds and* attitudes allowing for wind.

SYSCOM sets up communication with system variables by creating a NAMELIST table and reads 'semi-permanent' data changes from a file.

Execution of most routines during an initialisation pass is organised intrinsically by the SL1 model program (see section 5.2). Many routines may need to perform preliminary calculations. If their nature is such that they must be performed *once only* (otherwise errors will result) then a local flag must be created and the routine structured as in Fig 3a, with the flag set 'off' on entry (*eg* by a DATA statement). However, if there are no reasons why initial calculations should not be performed more than once there is a standard flag available in the system COMMON, JJCOMP, which takes the value 0 until the 'compute' button is pressed, and thereafter is 1 . In this case the structure in Fig 3b can be used.

4.2 Rotational motion

A diagram of the information flow is shown in Fig 4. Rotational motion is calculated first, principally to provide the direction cosines required by the translational motion.

)08

SACCROT calculates the three components of angular acceleration, in body axes, using equations (19), given total moments (supplied by the user) and inertia constants, equations (20), (calculated by SINIT).

SEULER calculates the rate of change of body attitude angles from the angular velocity components, using equations (1).

SDCOS calculates the nine direction cosines S_{11} etc from the three attitude angles ψ, θ, ϕ using equations (3).

4.3 Translational motion

A diagram of the information flow is shown in Fig 5.

SVELOC1 derives components of velocity, relative to the air, in earth axes using equations (9) and transforms to body axes, using equations (2).

SALFBET takes the body axes velocity components and calculates α, β, $\dot{\alpha}$, $\dot{\beta}$ using equations (12) and (13).

SACCLIN transforms total force components in body frame to inertial (earth) frame, using equations (7), then calculates the translational acceleration using equations (8). It may sometimes be desirable to separate the vertical from the two horizontal components so that vertical motion can be solved more frequently for better simulation of undercarriage dynamics.

SPATH calculates flight path angles from velocities in earth axes, using equations (15) and (16).

SVELOC2 calculates resultant airspeed, dynamic pressure, Mach number etc, using SATMOS for atmospheric properties. Some features are at present only valid for Mach number less than 1.0, as indicated in the routine.

4.4 Utility functions

Fig 6 shows the flow of information among the utility routines, all of which have something to do with the simulation environment, *eg* visual display, motion cues, wind etc.

STV calculates, from equations (22) to (25), positions and velocities to drive the TV visual system, and also handles the logic of belt positioning.

SILS calculates one or two segment ILS guidance beams.

SWIND controls the generation of turbulence and adds it to wind, modified by shear effects (if any).

SACCBOD calculates accelerations in body axes of the aircraft centre of gravity, from equations (26), and pilot station, accelerometers etc, from equations (27).

SATMOS calculates the properties (in ratio form) of the ICAO international standard atmosphere. It is an existing library subroutine and is not otherwise described in this Report.

A further utility routine, SCOUNT, not included in Fig 6, has nothing to do with the simulation of aircraft, but assists in the management of the calculation process, by setting an 'initialisation complete' flag, by keeping track of the sub-steps (if any) of the integration routine, and by picking up the current run number from the data logging system.

4.5 Units

With Aeronautics still retaining the use of such units as knots, it is difficult at present to make a wholehearted conversion to SI units. However, the equations set up are self-consistent so that a force in newtons acting on a mass in kilograms will produce an acceleration in m/s^2 and likewise, a force in pounds and mass in slugs will yield ft/s^2. The routines as set up are intrinsically in the feet, pound, second system but alteration of a few conversion constants (gravity, etc) and datum values (air density, etc), would enable the whole system to work in SI units.

5 OVERALL PROGRAM STRUCTURE

5.1 Introduction

The program which provides the general framework for all the system routines is written in the simulation language SL1 [14].

The basic structure (Fig 7) consists of an INITIAL region, for start-up calculations, a DYNAMIC region and a TERMINAL region. (The TERMINAL region, however, has no relevance to real-time simulation.) This is a common form for continuous system simulation languages, like SL1 or CSMP. The DYNAMIC region contains one or more DERIVATIVE sections. These DERIVATIVE sections, each of which can have its own integration algorithm and step size, contain the routines which are the core of a simulation. During real-time operation, the code generated from each DERIVATIVE section is executed repetitively to produce the desired solution as a function of time. A listing of an aircraft simulation program with one DERIVATIVE section, or loop, is given in Appendix C, and corresponding flow charts of the execution sequence are shown in Fig 8. The reason for using SL1

)08

is that two major facilities are provided by the language. These are synchronisation with real-time and a centralised integration scheme with a choice of five algorithms. *Further details are contained in the Reference* Manuals[14-16] and in Ref 1.

The SL1 program is translated into a series of Fortran modules. Communication is organised using Fortran labelled COMMON. To permit system COMMON to handle the main variables, however, some duplication of variables is necessary in order for SESAME routines to communicate with the integration operators.

This duplication is arranged through PROCEDURAL blocks, with the subterfuge that only those variables to be exchanged with the integration processes are contained in the argument list. Other variables thus remain invisible to the SL1 translator and so may be placed in system COMMON. If they were visible to the SL1 translator they would be placed in a separate COMMON area, no longer under control of the user.

For example, a PROCEDURAL block

```
PROCEDURAL (=VV)
      V = VV
END
```

picks up VV from the SL1 labelled COMMON and places its value in V . In the opposite direction

```
PROCEDURAL (VVDOT=)
      VVDOT = VDOT
END
```

transfers VDOT to VVDOT , for use in the integration statement

```
VV = INTEG (VVDOT, VVIC).
```

5.2 Initialisation

In the normal course of events, the code generated by the SL1 translator causes the DERIVATIVE section(s) to be executed just once at the end of INITIAL to calculate the initial values of all the derivatives. This has been augmented by explicit code (see near the 'END' of the INITIAL region in the SL1 listing in Appendix C) to force these calculations to be done twice. This is necessary because routines at the end of the sequence (such as SVELOC2) produce variables, such as dynamic pressure, needed early on. Once the whole system is running, there is no problem. This slight idiosyncracy needs to be remembered by the user. The other important point about initialisation is that at the start of a 008
new run, a *fresh copy* of the program is loaded into main store from a disc file,

so all parameter values are restored to their datum values. However, the 'retained changes' facility (see section 7.7) permits revised parameter values to be retained from run to run.

5.3 PV100 - operator dialogue

A set of programs and routines, collectively known as the parameter variation package, exists to enable the user to interact on-line with the parameters of the simulation. The principal item is PV100, the operator dialogue routine, which provides a wider range of facilities than the 'interpreter' inherent in SL1. PV100 enables the user, on-line and during any phase of a simulation flight, to access, display and amend any variable within the scope of the model program and its subroutines. Variables may be referred to by name (the principal function of routines SYSCOM and USERCOM is to provide tables of names for use by PV100), and in addition the contents of *any* memory location may also be displayed. A display may be in alternative styles, *eg* binary, hexadecimal, integer or 'real' format. Selected changes may be 'retained' to operate for subsequent runs, otherwise the change vanishes at the end of the current run. A readable memory dump may be printed, such as all the parameters in system COMMON. There is also a facility for forcing an 'automatic hold' or freeze if a variable falls within defined limits. A description of the full facilities is given elsewhere[17].

During initialisation there are four explicit calls to PV100 inserted at strategic points in the INITIAL region, as can be seen by inspection of the SL1 listing in Appendix C. These calls to PV100 are only executed if a switch (Desk Switch 1) on the simulator control desk is set on. These individual calls to PV100 may be identified by reference to the indicator ITP, as follows.

ITP = 1 After initial conditions or retained changes have been input, but before SINIT. Enables IC values to be altered, and other changes to be inserted before the main initialisation calculations of SINIT. In particular, if the frame time is to be altered, it should be done here, using FRAMET1, DELT1 for Loop 1 and similar variables for Loop 2 (if present).

ITP = 2 After SINIT but before the integrator IC values are actually set up.

ITP = 3 At end of INITIAL but before analogue to digital converter (ADC) read and DERIVATIVE section(s) initialised.

ITP = 4 After DERIVATIVE section(s) initialised and ADC read. DAC values are calculated but not yet set.

6 COMMUNICATION

6.1 Introduction

Communication among all the routines which comprise the aircraft model program is achieved by Fortran labelled COMMON. Labelled COMMON designates a block of contiguous memory locations which may be accessed by any subroutine in which the COMMON block is declared. This technique provides flexibility and also avoids the time penalty associated with using arguments in subroutine calls.

Two large blocks of labelled COMMON are employed. One, named SYSTEM, is an inherent part of the standard subroutines and contains all the state variables and other parameters common to any aircraft simulation. This is of fixed size and has its variables in predefined locations. The other, named USER, is set up, in terms of size and content, by the user to handle variables specific to his particular simulation, *eg* the aerodynamic force coefficients, engine forces etc.

6.2 System COMMON

This is defined as

```
COMMON / SYSTEM / A(1000), L(400)
```

where the array A contains real variables and L contains integer variables. An index to all the variables, in numerical and alphabetical order, is provided in Appendices D and E.

If the user wishes to pick up the current value of angle of attack (say) he needs to include in his routine

```
COMMON / SYSTEM / A(1000), L(400)
EQUIVALENCE (ALFAD , A(112))
```

and then ALFAD can be used freely in his routine. System variables must *not* be altered by the user.

6.3 User COMMON

This is defined for example as

```
COMMON / USER / B(1000), M(100)
```

where the array B is for real variables and M for integer variables. Names and sizes of these arrays are defined by the user, but the COMMON name USER must be employed. Sizes should be kept as small as possible, in order to avoid sterilising areas of core unnecessarily. However, if the size is too small, the user will have to increase it frequently, with consequent work updating *all* routines in which the COMMON statement appears.

The user will choose his own variable names and allocate space in the COMMON arrays. This will normally be done by EQUIVALENCE techniques, for flexibility. (Routines could, in fact, be created and checked in the first place without considering inter-routine communication.)

6.4 Documentation aids

A utility program has been created, called COMMLIST, to assist the user in keeping track of his variable names and locations. As he creates a variable, the user punches a card defining the variable name, its meaning and the routine in which it is calculated. This is similar to a scheme described by Bean[18]. The program COMMLIST can then produce an index of names in alphabetical order, of locations in numerical order or of names according to the routine in which they are created or used. A brief guide to the program COMMLIST is given in Appendix F. The lists in Appendices D and E, G and H have been produced by this program.

6.5 Access to variables at run-time

In order to be able to access his variables at run-time, to inspect and change values via PV100[17], the user must create a NAMELIST table of his variable names. This is achieved in the routine USERCOM, which must be written by the user. The bulk of this routine will be COMMON (or EQUIVALENCE) statements, referencing the variables by name. Its structure will be identical to the routine SYSCOM which does the same job for system variables. A listing of SYSCOM is included in Appendix B.

6.6 Hybrid input/output (ADC, DAC)

6.6.1 Introduction

Hybrid input/output, that is analogue to digital conversion (ADC) and digital to analogue conversion (DAC), is integrated into SESAME so that the user merely has to define entries in a set of tables incorporated in system COMMON. Conversion and scaling then occurs automatically. The general form of conversion required is of the form

$$y = ax + b$$

where a is a scale factor and b is a bias. For ADC, x would be a scaled value and y a value in 'engineering' units whereas, for DAC, x would be in engineering units and y a value scaled appropriately for output. The present implementation does not include any bias.

Data transfer in both directions employs a special piece of hardware known as the DMS12 (Direct Memory Sub System) which is a processor capable of operating in parallel with the computer's central processor. Use of this device provides for an element of parallel operation which is exploited in the software to save execution time.

Overall, the software scheme offers the user convenience - because he only has to set up data tables; flexibility - because the tables can be changed on line; and speed - because the hardware is fully exploited. The operating routines, coded in assembler, are not described in this Report.

6.6.2 Analogue to digital conversion

Analogue to digital conversion, and scaling, are executed prior to entry into the Derivative Section of the SL1 model program, where the dynamic equations are solved. The main scaling calculation performed is of the form

$$YADC(I) = ADC(NADC(I)) * AADC(I)$$

where ADC is the array of raw converted values, nominally in the range ± 1.0, held in the SL1 labelled COMMON Z99999.

NADC is the array of ADC channel numbers

AADC is the array of scaling factors, set up by the user

YADC is the array of input values in engineering units (degrees etc) ready for use.

All 32 channels are read in raw form into the array ADC, but scaling and transfer to YADC stops when the first zero element in NADC is reached. Because of the parallel nature of the hardware (DMS12, mentioned above) scaling is begun as soon as the first raw converted value is available. Conversion of further channels then proceeds concurrently with scaling. To avoid the danger of scaling a channel before it has been converted, it is important, therefore, that ADC channels are kept in numeric order and are used, as far as possible, without large gaps. For example, to use channels 1, 3, 7, 9 is better than 1, 17, 24, 30. To use 1, 2, 3, 4 is best of all. Thus, for simplicity, NADC(I) = I.

The data structure being in table form allows considerable operational flexibility. For example, in the event of failure of ADC channel 2, the variable being input on this channel can be redirected to channel 9 (say) simply by repatching on the analogue computer and by changing the contents of NADC(2) from 2 to 9 via the keyboard. The scaling AADC(2) would not need to be altered. The 008
necessary changes can all be achieved on-line and no program changes are necessary. However, the warning of the previous paragraph should not be overlooked.

At present 32 ADC channels are available and supported by software. Provision has been made for enlargement, however, by reserving 64 contiguous locations for each of the arrays AADC, YADC, NADC in the COMMON storage area.

The SL1 model program includes in its source code (Appendix C) a sequence of the form

```
PROCEDURAL (=ADC(1))
END
```

in order to ensure that the key driving routine Z9991 is called. No special code has to be created by the user.

The user does, of course, wish to use the result of the conversion and scaling processes, as held in YADC. To do so he may either use an element of YADC directly, as in the following example

```
CL  =  CLETA*YADC(7) + .......
```

or may give the element of YADC his own names via the usual EQUIVALENCE declarations *viz*

```
DIMENSION YADC(32)
EQUIVALENCE (YADC(1), A(401))
EQUIVALENCE (ETAD   , YADC(7))
```

and then

```
CL  =  CLETA * ETAD + ......
```

To make his variable names accessible via PV100, the user could include them in subroutine USERCOM.

6.6.3 Digital to analogue conversion

Digital to analogue conversion occurs once per frame, *after* the main integration loop. The DAC software scales output variables to be in the settable range for DACs and DCUs (otherwise known as vari-DACs).

In principle, the same calculation technique is adopted as for ADC. The special difficulty for variables to be output to DACs is that they must be selected from any location in the system or user COMMON arrays (A or B) and must be routed either to proper, four-quadrant DACs (the first 32) or to two-quadrant vari-DACs (the next 16). An additional requirement is that it should be possible to set DACs from either of two loops. (This description refers for completeness to two loops. For the moment, however, only Loop 1 is operational.)

To achieve these objectives, data for four arrays must be defined by the user.

ADAC array of scaling factors. A variable in engineering units is multiplied by the scaling factor in the appropriate element to give a result in the desired range (nominally ±1.0).

NADC is the array of pointers defining which variables are to be converted. The pointers may refer to the 'A' or 'B' arrays in any sequence. Distinction between the source arrays is controlled by arrays L1DAC, L2DAC below.

L1DAC defines the source of data for each channel to be converted in Loop 1.

L2DAC defines the source of data for each channel to be converted in Loop 2. Each element of L1DAC, L2DAC can take the value 0, 1 or 2: 1 means that the data for scaling is to be taken from array A and 2 that it comes from array B. If 0 is set, no scaling is performed.

As an example, if NDAC(6) = 112, L1DAC(6) = 1, ADAC(6) = 0.1 then DAC channel 6 is to be used to output, at the end of Loop 1, variable A(112), *ie* ALFAD, angle of attack in degrees, scaled to ±10.0.

This structure provides flexibility by enabling changes to be introduced on-line. For example, by changing NDAC(6) from 112 to 241 (say), and changing the scaling ADAC(6) to suit, a different variable from the same array can be output on a given DAC. The source array (A or B) can also be easily changed, via L1DAC.

Operational considerations

(1) Of the 48 DAC channels available, the first 32 are true DACs, while the rest are only two-quadrant devices. The user needs to remember this when allocating channels, so that 33-48 are only used for variables which do not change sign, *eg* airspeed. Provision has been made for expansion to 64 channels by reserving extra space for ADAC, NDAC in the COMMON area.

(2) For those DAC channels not used, the corresponding elements of ADAC should be set to zero.

(3) A set of default hardware addresses associated with channel numbers is defined in the table IDACAD. DAC addresses can be reassigned during initialisation.

(4) Timing tests have given the following results:

Convert and scale 32 ADC channels	1.2 ms
Scale and convert 48 DAC channels	3.8 ms .

All channels are converted, but only those specified are scaled and stored, so that time can be saved if fewer channels are needed. 008

6.6.4 User summary of system COMMON relevant to ADC/DAC

A	301 364		401 464		501 564
	1 AADC 64		1 YADC 64		1 ADAC 64

A System COMMON array (real)

AADC* Array of scaling factors for ADC

YADC Array of converted variables

ADAC* Array of scaling factors for DAC

L	11 74	82 144		305 352	353 400
	1 NDAC 64	1 NADC 64		1 L1DAC 48	1 L2DAC 48

L System COMMON array (integer)

NDAC* Array of pointers defining variables in 'A' array and user's 'B' array to be converted

NADC* Array of ADC channel numbers

L1DAC* Array of DAC source pointers for Loop 1

L2DAC* Array of DAC source pointers for Loop 2

6.7 Hybrid input/output (discretes)

6.7.1 Introduction

Discretes are single-bit logic lines enabling on/off information to be communicated into and out of the computer. These lines are read and set automatically and so are conveniently available to the user.

6.7.2 Summary of lines available

A number of lines are available in each direction

(a) between the digital computer (Sigma 8) and the analogue computer (AD4) and

(b) between the digital computer and other external hardware, such as the simulator cockpits and the control desk. The numbers and types are summarised in the table below, together with the relevant driving routine.

* These items must have values supplied by the user. It is recommended that, for convenience, all such data be located in one routine, such as OUTSR, or a self-contained 'information' routine.

008

	Type	Location	Number	Direction	Subroutine
(1)	AD4 sense lines	AD4 logic patch panel	16	To Sigma 8	READSLR
(2)	AD4 control lines	AD4 logic patch panel	16	From Sigma 8	SETCLR
(3)	Patchable sense lines	Rack N logic patch panel	32	To Sigma 8	READSCR
(4)	Desk switches	Bay 5 of control desk	32	To Sigma 8	READSCR
(5)	Relay change-overs	Rack N logic patch panel	32	From Sigma 8	SETDSCR

The driver routines READSLR etc are themselves executed by one of two supervisor routines DSCRT1 and DSCRT2. DSCRT2 is only relevant if the model program is structured to have two iteration loops.

6.7.3 Software interface

Discrete information to be input or output is communicated via the labelled COMMON system, *viz*

COMMON/SYSTEM/A(1000), L(400).

The basic routines which handle the hardware are called implicitly and convert from bit patterns in 32-bit words to arrays of integers, as defined below. Each integer can take the value 0 or 1. Thus no coding or decoding (packing or unpacking) of bits is necessary by the user.

The layout of the relevant parts of L is

L	150	151 ... 166	170	171 ... 186	190	191 ... 222	223 ... 254	270	271 ... 302
	IAD4SL	1 ISLR 16	IAD4CL	1 ICLR 16	IDSCFL	1 IP 32	1 IDS 32	IPCOFL	1 ICO 32

where ISLR(16) is the AD4 sense line array, ISLR(1) corresponding to DGS00 on the AD4 patch panel (remember the AD4 is marked in octal!)

ICLR(16) is the AD4 control line array, ICLR(1) corresponding to DGC00 on the AD4 patch panel

IP(32) is the array of patchable sense lines

IDS(32) is the array of desk switches

ICO(32) is the array of relay change-overs

IAD4SL, IAD4CL, IDSCFL, IPCOFL are flags, defined in detail in section 7.4.1 below, which control in which iteration loop, if any, the discretes routines are executed.

All these names are defined in SYSCOM, to be accessible by PV100. For the user to obtain access to ISLR(3), for example, he will include in his routine

```
                  COMMON/SYSTEM/A(1000), L(400)
and either        EQUIVALENCE (ISLR(1), L(151))
                  DIMENSION ISLR(16)
or                EQUIVALENCE (ISLR(3), L(153))
```

The former is more general. Discrete information is then available for use in such constructs as

```
IF (ISLR(3).EQ.1) CALL XYZ
```

6.7.4 Dedicated functions

Some of these discrete lines are dedicated to certain functions within SESAME, as summarised in the following list.

Line	Purpose	Routine
ISLR(4)	Slew forward	STV
ISLR(5)	Slew back	STV
ICLR(16)	TV Desync	STV
IDS(1)	PV100 select	SL1
IDS(5)	TV cycle	STV
IDS(6)	X TV reset	STV
IDS(7)	Y TV reset	STV
IDS(8)	Turbulence	SWIND

7 USER'S GUIDE

7.1 What the system expects from the user

The user is required to provide the three components of total force (FTX, FTY, FTZ) and the three components of total moment (XLLTOT, XMMTOT, XNNTOT), all in body axes. Thereafter the appropriate integrations, resolutions etc occur, with the system providing back to the user all the state variables from which the forces etc are generated. Various constants must also be provided. These are described in section 7.3.

7.2 Typical user routines

The user is required to create the following set of routines, which may perform some or all of the tasks indicated.

)08

'Input'

CONTROLS takes the raw control inputs from the pilot, applies any non-linear gearings, computes autostabiliser contributions and finishes up with the total control surface deflections, for use by the aerodynamics routines. It may also organise the computation of engine performance, including dynamics, and total thrust, momentum drag etc.

'Calculation loop'

TOTF computes aerodynamic force coefficients etc, finishing up with three components of total force from all sources.

TOTM computes aerodynamic moment coefficients etc, finishing up with three components of total moment from all sources.

'Output'

OUTSR computes miscellaneous functions, *eg* scalings of non-linear instruments, navigation and guidance and handles data logging.

'Initial'

USERCOM contains the names of all the variables the user may wish to access via PV100, and also reads semi-permanent data changes from a file.

* These routines are actually called at SL1 level. All are executed in a DERIVATIVE section, except USERCOM, which is executed in INITIAL.

* All these routines are likely to be umbrella routines, in the sense that they are likely merely to organise the calling of other, more detailed, routines. For example, TOTF may call a routine to compute all ground reactions, tyre forces etc.

* Ideally, TOTF and TOTM should be kept separate, so that they can, if necessary, be executed in different loops at different frame times.

* Routines in the calculation loop may be executed up to four times per frame, depending on the integration technique employed, *eg* Trapezoidal, Runge-Kutta fourth order. All other routines are only executed once per frame (see section 5.4).

* Each routine should, as far as possible, contain its own constants.

* The method for communication of variables other than by lengthy argument lists has been described in section 6.

7.3 Data the user must provide

In addition to the primary forces and moments (FTX, XLLTOT, etc) and initial values (section 7.5), the user must provide values for a number of constants and parameters. These fall into such categories as basic data, hybrid input/output and external environment. They are described briefly below and listed in full in Appendices G and H. Typically the values will be provided by DATA statements

in appropriate subroutines (*eg* TOTM, TOTF), unless default values are already provided, in which case new values should be provided in the SYSCHNG file (section 7.7). Zero values should be set up if the parameter is not relevant.

Aircraft basic data

Moments of inertia	XIX, XIY, XIZ, XIZX
Centre of gravity	XCGREF, ZCGREF, ZCG
Span, wing area etc	SPAN, SWREF, STAIL, XLTAIL, CREF
Pilot location	XP, ZP
Coordinates of slip ball	X1, Y1, Z1
Coordinates of g meter	X2, Y2, Z2
Coordinates of accelerometer (AX3)	X3, Y3, Z3
Coordinates of accelerometer (AY4)	X4, Y4, Z4
Coordinates of accelerometer (AZ5)	X5, Y5, Z5

Timing

Frame time	FRAMET1, FRAMET2
Step size	DELT1, DELT2

Hybrid input/output

Scaling factors for analogue to digital conversion etc, (for full details refer to section 6.6)	AADC, ADAC L1DAC, L2DAC, NDAC, NADC
Output discretes (see section 6.7)	ICLR, ICO

External environment

Turbulence rms	USIG, VSIG, WSIG
Turbulence character	NG, SFRACG, SRDECAY
Ship speed	VSHIPKT

Additional integer parameters which can take one of several specific values and thereby select alternative functions are described in the next section.

7.4 Options available

A number of choices may be made by software flags or hardware switches. These are summarised below, with default values where relevant.

7.4.1 Software flags

(a) Variation of atmospheric properties

KISA	default 0 set in SINIT
0	constant sea level conditions at all times
1	standard atmosphere as function of current height
-1	constant properties appropriate to initial height (HIC).

008

FOR KISA = 1, SATMOS is called within SVELOC2 to calculate the new properties. Otherwise, constant values are set up within SINIT.

(b) Selection of constants for TV belt in use

NTVB	no default
1	700:1 model
2	2000:1 model
3	5000:1 model

(c) ILS on/off

LSILS	no default
0	ILS off
1	ILS on

(d) Type of ILS beam

ILSFLG	no default, value obtained from XILSFLG (see section 7.5.2)
1	3° straight beam
2	6° straight beam
3	6° changing to 3° at height HKINK (qv)

(e) Discretes in and out (see also section 6.7)

IAD4SL	AD4 sense lines (default 1 set in DSCRT1)
IAD4CL	AD4 control lines (default 1 set in DSCRT1)
IDSCFL	discretes in to Sigma (default 1 set in DSCRT1)
IPCOFL	discretes out from Sigma (default 0 set in DSCRT1)
0	do not execute at all
1	execute in DSCRT1, *ie* in fast loop
2	execute in DSCRT2, *ie* in slow loop (if any)

(f) Wind shear

ISHR	default 1, set in SINIT
1	no shear
2	logarithmic profile
3	linear profile

The value of ISHR is obtained from XISHR set up in the Initial Conditions file.

(g) Control of random number generation for turbulence

LSEED default 0, set in SWIND

0	seeds for random number generation derived from time of day
1	constant seeds employed, enabling a repeatable turbulence sequence to be obtained

008

7.4.2 External switches

Desk switch	Name	Purpose when set on
1		Calls PV100 during INITIAL
5	LCYCLE	TV belt cycles continually
6	LXTVIC	Resets TV x position
7	LYTVIC	Resets TV y position
8	LTURB	Turbulence
TV slew	LFWD	TV belt (and x position) moves forward at maximum rate
TV slew	LBACK	TV belt (and x position) moves backward at maximum rate

7.5 Initial conditions and initial values

7.5.1 Initial conditions

The standard SL1 model embodies 12 integrations, each of which requires an initial condition. These are given in the table below as 'SL1 names'. Many of these variables do not form a natural or convenient set for the user, who is more interested, for example, in specifying an initial airspeed rather than components of speed relative to the ground, or an initial angle of attack rather than pitch attitude. Hence SINIT performs various calculations to transform a user's initial values into those that the system requires.

One particular feature is that, regardless of the wind, the ground speed components are derived implicitly to satisfy the specified initial *heading* PSIDIC so that any datum wind or wind shear conditions are not revealed inadvertently to the pilot. More detail is given in the description of the SINIT routine in Appendix A.

SL1 names	Source	Definition
VVKNIC	Calculation	Velocity comp rel to ground, north
VVKEIC	Calculation	Velocity comp rel to ground, east
VVKDIC	Calculation	Velocity comp rel to ground, down
SXIC	XIC	X position
SYIC	YIC	Y position
SHIC	HIC	Height
PPIC	PDIC, default zero	Rate of roll, body axes
QQIC	QDIC, default zero	Rate of pitch, body axes
RRIC	RDIC, default zero	Rate of yaw, body axes
PPHIC	PHIDIC, default zero	Roll attitude angle
TTHIC	Calculation	Pitch attitude angle
PPSIC	PSIDIC	Heading angle

008

7.5.2 Initial values

Initial values are parameters relevant to the current 'run' and are not in general true initial conditions for integration. However, those initial conditions listed above that are not calculated are obtained from a set of initial values created by the user in a file. This file is read, during the INITIAL phase, by the routine RDICFILE and its contents stored in a 20-element array (A(191) - A(210)) which is part of system COMMON.

At present only 14 elements are actually used (as defined by the value of NVALS). These are

Name	Element	Definition	Units
BETADIC	A(191)	Sideslip angle	degrees
ALFADIC	A(192)	Angle of attack	degrees
GAMDIC	A(193)	Climb angle	degrees
VKTIC	A(194)	Airspeed	knots
XIC	A(195)	X position	feet
YIC	A(196)	Y position	feet
HIC	A(197)	Height	feet
W	A(198)	Aircraft weight	pound
XCG	A(199)	Aircraft centre of gravity	fraction of ref chord
XILSFLG	A(200)	ILS flag (actually used in integer form, ILSFLG)	
XISHR	A(201)	Wind shear flag (actually used in integer form, ISHR)	
VWKTØ	A(202)	Datum wind speed	knots
PSIWD	A(203)	Wind direction	degrees
PSIDIC	A(204)	Aircraft heading	degrees
Spare	A(205) - A(210)		

Additional items could be read by increasing NVALS, *eg* by inputting a new value from the semi-permanent changes file (see section 7.7) read in SYSCOM.

7.6 Modification status

To keep track of the modification status of each routine, it is recommended that the user allocate an identifier per routine and assign a value to identify the version. For example, with the routine TOTF

```
DATA MTOTF / 1 /
```

to indicate version 1, or

DATA MTOTF / 171176/

to indicate the date of the current version. These values should be changed whenever the routine is modified. By placing these status identifiers in user COMMON, the current versions of each routine can be readily checked on-line.

This technique has been adopted for all the SESAME routines. The name of the version identifier is the routine name preceded by K , *eg* for routine SALFBET, the identifier is KSALFBET and contains the date in integer form. These identifiers are held in system COMMON (see Appendix E).

Identification of a new version of the executable program, or load module, is desirable. The SL1 model program has a version number (IVERSION) for its *source* code, but this is not changed if a new load module is built without changing the SL1 source. Identifying individual routines as above will help but a global load module identifier could also be created in a 'history' routine, the only purpose of which is to set a version number for the load module, *eg*

DATA LHARRIER / 240776 /

and perhaps also to include a modification history in 'comment' form.

A recent modification to the loader utility in the RBM operating system now places the date and time of load module creation into the program header. This information can be inspected from the keyboard.

7.7 Changes to parameters

It is possible to introduce lasting parameter changes into a completed and working aircraft model program without recoding.

On-line changes can be achieved from a keyboard, during any mode of a simulation, through the program PV100[17]. The user has the option, at the time of making the changes, of declaring that the changes be 'retained', in which case they are automatically copied into a file and read back again (by PV300) at the start of each subsequent run. These changes are initially name-orientated, but PV300 relies on absolute address so that if alternative aircraft model programs are used, the retained changes file needs to be reset.

Semi-permanent changes to a user's parameters held in USER COMMON may be achieved by creating (from cards) a file called USERCHNG which is read each time the routine USERCOM is executed. Changes to parameters in SYSTEM COMMON may be achieved by a similar process, via the file SYSCHNG. All these changes work only by parameter name, as explained in the description of SYSCOM in Appendix A.

)08

7.8 Operational considerations

The binary versions of all SESAME routines, in relocatable form (ROM), are held in files on disc and may be incorporated when building, via OLOAD, the final executable program. To ensure that the size of user defined COMMON is correctly set up, it is essential that a user ROM containing the definition of USER COMMON be loaded *before* SDAC.

If data logging is in use, the correct run number may be obtained by inspecting the variable NRUN.

8 CONCLUSIONS

A system of equations and associated computer subroutines has been developed to make it easy to create an aircraft model program for real-time flight simulation. Several simulations have been completed and it has been found to be very useful. The objectives of the scheme have been achieved. In particular, the resulting model program is more efficient during execution in terms of time required and the modular structure has also given the desired benefits at the program development stage.

The facilities of SESAME are described as they exist at the time of publication of this Report. Extensions and enhancements may easily be added as new needs arise.

Furthermore, the routines as described in this Report are not specific either to real-time simulation or to the computer on which they have been developed. They are written in Fortran and, supplemented by appropriate integration techniques, could be used on other computers for general purpose calculations of aircraft dynamics in six degrees of freedom.

Appendix A

DESCRIPTION OF INDIVIDUAL SESAME ROUTINES

Section	Routine	Source file	Binary file (ROM)
A.1	SACCBOD	SACBS	SACBR
A.2	SACCLIN	SACLS	SACLR
A.3	SACCROT	SACRS	SACRR
A.4	SALFBET	SALFS	SALFR
A.5	SCOUNT	SCNTS	SCNTR
A.6	SDCOS	SDCSS	SDCSR
A.7	SEULER	SEULS	SEULR
A.8	SILS	SILSS	SILSR
A.9	SINIT	SINTS	SINTR
A.10	SPATH	SPTHS	SPTHR
A.11	STV	STVVS	STVVR
A.12	SVELOC1	SVL1S	SVL1R
A.13	SVELOC2	SVL2S	SVL2R
A.14	SWIND	SWNDS	SWNDR
A.15	SYSCOM	SYSCS	SYSCR
A.16	DSCRT1	SCR1S	SCR1R
A.16	DSCRT2	SCR2S	SCR2R
A.17	ISDSCR	ISDCS	ISDCR

)08

A.1 Subroutine SACCBOD

A.1.1 Purpose — Calculates accelerations, in body axes, of aircraft centre of gravity (cg), pilot and various accelerometer locations.

A.1.2 Call — CALL SACCBOD

A.1.3 Inputs
P,Q,R
PDOT, QDOT, RDOT
RG, W
FTX, FTY, FTZ
S13, S23, S33
XP, ZP
X1, Y1, Z1; X2, Y2, Z2; X3, Y3, Z3; X4, Y4, Z4; X5, Y5, Z5

A.1.4 Outputs
AXCG, AYCG, AZCG
AXACG, AYACG, AZACG
AXP, AYP, AZP
AZAP, AZCG1
AY1, AZ2, AX3, AY4, AZ5

A.1.5 Data required
XP, ZP
X1, Y1, Z1; X2, Y2, Z2; X3, Y3, Z3; X4, Y4, Z4; X5, Y5, Z5

A.1.6 Description (NB all accelerations in units of g) - see also section 3.9

A.1.6.1 Specific accelerations at aircraft centre of gravity (as measured by an accelerometer).

$$a_{xcg} = F_{TX}/W$$

$$a_{ycg} = F_{TY}/W$$

$$a_{zcg} = F_{TZ}/W$$

Note that in steady level flight, with $\theta = 0$, a 'normal' accelerometer at the centre of gravity registers -1 g and in a pull-up, for example, registers -1.5 g.

A.1.6.2 Absolute accelerations at aircraft centre of gravity

$$a_{xacg} = a_{xcg} + S_{13}$$

$$a_{yacg} = a_{ycg} + S_{23}$$

$$a_{zacg} = a_{zcg} + S_{33}$$

S_{13}, S_{23}, S_{33} are direction cosines, *eg* $S_{13} = - \sin\theta$.

A.1.6.3 Accelerations at pilot

$$a_{xp} = a_{xcg} - \left[x_p(q^2 + r^2) - z_p(pr + \dot{q})\right]/g$$

$$a_{yp} = a_{ycg} + \left[x_p(pq + \dot{r}) + z_p(qr - \dot{p})\right]/g$$

$$a_{zp} = a_{zcg} + \left[x_p(pr - \dot{q}) - z_p(p^2 + q^2)\right]/g$$

A.1.6.4 Absolute normal acceleration at pilot (or 'manoeuvre g' used for motion cues)

$$a_{zap} = a_{zp} + S_{33}$$

A.1.6.5 Acceleration relative to 1 g for recording

$$a_{zcg1} = a_{zcg} + 1.0$$

A.1.6.6 Acceleration for slip ball (located at x_1, y_1, z_1)

$$a_{y_1} = a_{ycg} + \left[x_1(pq - \dot{r}) - y_1(p^2 + r^2) + z_1(qr - \dot{p})\right]/g$$

A.1.6.7 Accelerations for g meter (located at x_2, y_2, z_2)

$$a_{z_2} = a_{zcg} + \left[x_2(pr - \dot{q}) + y_2(qr - \dot{p}) - z_2(p^2 + q^2)\right]/g$$

A.1.6.8 Accelerations for three independent accelerometers

$$a_{x_3} = a_{xcg} - \left[x_3(p^2 + r^2) - y_3(pq - \dot{r}) - z_3(pr + \dot{q})\right]/g$$

$$a_{y_4} = a_{ycg} + \left[x_4(pq + \dot{r}) - y_4(p^2 + r^2) + z_4(qr - \dot{p})\right]/g$$

$$a_{z_5} = a_{zcg} + \left[x_5(pr - \dot{q}) + y_5(qr + \dot{p}) - z_5(p^2 + q^2)\right]/g$$

A.1.7 Initialisation

Nil

A.1.8 Subroutines used

Nil

A.1.9 **Remarks**

)08

A.1.9.1 See Etkin[9] (Dynamics of Atmospheric Flight) pp 122-124 for derivation of acceleration at an arbitrary point.

A.1.9.2 'Flat earth' is assumed.

A.2 Subroutine SACCLIN

A.2.1 Purpose To compute linear accelerations in earth axes

A.2.2 Call CALL SACCLIN

A.2.3 Inputs FTX, FTY, FTZ
RXMASS, G
S_{11} - S_{33}

A.2.4 Outputs FTN, FTE, FTD
VKNDOT, VKEDOT, VKDDOT

A.2.5 Description

A.2.5.1 Transforms total forces in body axes (FTX, FTY, FTZ) to earth axes (FTN, FTE, FTD), using direction cosines S_{11} etc.

$$F_{TN} = S_{11}F_{TX} + S_{21}F_{TY} + S_{31}F_{TZ}$$

$$F_{TE} = S_{12}F_{TX} + S_{22}F_{TY} + S_{32}F_{TZ}$$

$$F_{TD} = S_{13}F_{TX} + S_{23}F_{TY} + S_{33}F_{TZ}$$

A.2.5.2 Calculates accelerations in earth axes

$$\dot{V}_{KN} = F_{TN}/m$$

$$\dot{V}_{KE} = F_{TE}/m$$

$$\dot{V}_{KD} = F_{TD}/m + g$$

A.2.6 Initialisation

Nil

A.2.7 Subroutines used

Nil

A.2.8 Remarks

User must supply total forces in body axes (FTX, FTY, FTZ).

A.3 Subroutine SACCROT

A.3.1 Purpose Calculates angular accelerations in body axes

A.3.2 Call CALL SACCROT

A.3.3 Inputs XLLTOT, XMMTOT, XNNTOT
P, Q, R
CI1 - CI10

A.3.4 Outputs PDOT, QDOT, RDOT

A.3.5 Description

Given total moments, angular velocity components and inertia coefficients, calculates angular acceleration components in body axes.

$$\dot{p} = CI_1L + CI_2N + CI_3pq + CI_4qr$$

$$\dot{q} = CI_5M + CI_6(r^2 - p^2) + CI_7rp$$

$$\dot{r} = CI_8N + CI_2L + CI_9pq + CI_{10}qr$$

A.3.6 Initialisation

Nil

A.3.7 Subroutines used

Nil

A.3.8 Remarks

A.3.8.1 CI1 etc are calculated in SINIT from inertia information supplied by the user.

A.3.8.2 The user must provide the total moments in body axes (XLLTOT, XMMTOT, XNNTOT).

A.3.8.3 An xz plane of symmetry is assumed.

A.4 Subroutine SALFBET

A.4.1 Purpose Calculates angles of attack and sideslip

A.4.2 Call CALL SALFBET(DT)

A.4.3 Inputs UB, VB, WB, RADTOD

A.4.4 Outputs ALFAR, ALFAD, SALFA, CALFA, ALFADOT,
BETAR, BETAD, SBETA, CBETA, BETADOT

A.4.5 Argument DT is frame time of loop in which SALFBET is used.

A.4.6 Description

008 A.4.6.1 Given the body axes components of airspeed, calculates α and β from (see Hopkin[4] section 6.2)

$\alpha = \tan^{-1}$ WB/UB, with α taking the sign of WB and being in the range $-\pi$ to $+\pi$.

$\beta = \tan^{-1}$ VB/(UB^2 + WB^2)$^{\frac{1}{2}}$
being in the range $-\pi/2$ to $+\pi/2$, and taking the sign of VB.

A.4.6.2 Calculates the sine and cosine of α and β.

A.4.6.3 Converts α and β to degrees and stores.

A.4.6.4 Estimates the rate of change of the incidence angles ($\dot{\alpha}$ and $\dot{\beta}$) from the present and two preceding values, *eg*

$$\dot{a} = (a_2 - 4a_1 + 3a)/(2DT)$$

where DT is the time step.

A.4.7 Initialisation

During the first entry to the routine the two preceding values of α and β used in estimating $\dot{\alpha}$, $\dot{\beta}$ are set to the current values of α and β, thus making $\dot{\alpha} = \dot{\beta} = 0.0$ for two time steps after 'compute'.

A.4.8 Subroutines used

Library mathematical functions.

A.4.9 Remarks

For UB = WB = 0.0, $\alpha = 0.0$.
If VB is zero as well, then $\beta = 0.0$.

A.5	Subroutine	SCOUNT
A.5.1	Purpose	Resets pass count for derivative section
A.5.2	Call	CALL SCOUNT
A.5.3	Inputs	JCOMP (from COMMON 'CONTROL')
A.5.4	Outputs	NCPASS, JJCOMP, NRUN

A.5.5 Description

A.5.5.1 Resets pass count NCPASS to zero. NCPASS is used in the derivative section to keep track of sub-steps of integration routine.

A.5.5.2 Picks up JCOMP from COMMON 'CONTROL' and reassigns to JJCOMP in COMMON 'SYSTEM' to enable JJCOMP to be used as an 'initialisation complete' flag. JCOMP (and hence JJCOMP) gets set to 1 once 'compute' is pressed.

A.5.5.3 Calls DLRUN to pick up current run number held in mailbox.

A.5.6 Initialisation

Nil

A.5.7 Subroutines used

DLRUN

A.5.8 Remarks

A.5.8.1 Since DLRUN accesses the foreground mailbox, any program using SCOUNT or DLRUN must be run in the foreground.

A.5.8.2 Run number is only obtained if data logging is *not* inhibited.

A.5.8.3 Because SCOUNT is not called during initialisation, the run number is not obtained until after 'compute' is pressed.

A.6	Subroutine	SDCOS
A.6.1	Purpose	Calculates direction cosines from Euler angles
A.6.2	Call	CALL SDCOS
A.6.3	Inputs	PSIR, THETAR, PHIR, RADTOD
A.6.4	Outputs	PSID, THETAD, PHID, SPSI, CPSI, STHETA, CTHETA, SPHI, CPHI

A.6.5 Description

A.6.5.1 Calculates sines and cosines of the three Euler angles

ψ - heading 0-360^{o} (or ± 180)

θ - pitch $\pm 90^{o}$

ϕ - bank $\pm 180^{o}$

A.6.5.2 Calculates nine direction cosines used in transformation from earth to body axes and vice versa (see Hopkin[4] section 5.6).

$$
\begin{array}{lcl}
S11 & = & \cos\theta\cos\psi & = & \ell_1 \\
S12 & = & \cos\theta\sin\psi & = & \ell_2 \\
S13 & = & -\sin\theta & = & \ell_3 \\
S21 & = & \sin\phi\sin\theta\cos\psi - \cos\phi\sin\psi & = & m_1 \\
S22 & = & \sin\phi\sin\theta\sin\psi + \cos\phi\cos\psi & = & m_2 \\
S23 & = & \sin\phi\cos\theta & = & m_3 \\
S31 & = & \cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi & = & n_1 \\
S32 & = & \cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi & = & n_2 \\
S33 & = & \cos\phi\cos\theta & = & n_3
\end{array}
$$

A.6.5.3 Calculates $\sec\theta$ and $\tan\theta$.

A.6.5.4 Converts input angles (ψ, θ, ϕ) from radians to degrees and stores.

A.6.6 Initialisation

Nil

A.6.7 Subroutines used

Library mathematical functions, *eg* sine, cosine

A.6.8 Remarks

Makes no provision yet for $\theta = 90^{\circ}$, nor for large angles, *ie* multiple turns.

A.7 Subroutine SEULER

A.7.1 Purpose Calculates Euler attitude rates.

A.7.2 Call CALL SEULER

A.7.3 Inputs P, Q, R
SPHI, CPHI, TANTHT, SECTHT
RADTOD

A.7.4 Outputs PHIDT, THETDT, PSIDT
PD, QD, RD

A.7.5 Description

A.7.5.1 Calculates Euler attitude rates from body rates and Euler angles

$$\dot{\phi} = p + (q \sin\phi + r\cos\phi)\tan\theta$$

$$\dot{\theta} = q\cos\phi - r\sin\phi$$

$$\dot{\psi} = (q\sin\phi + r\cos\phi)\sec\theta$$

A.7.5.2 Converts body rates to degrees/s and stores.

A.7.6 Initialisation

Nil

A.7.7 Subroutines used

Library mathematical functions.

A.8 Subroutine SILS

A.8.1 Purpose To calculate glide slope and localiser indications for conventional instrument landing system (ILS). 008

A.8.2 Call — CALL SILS

A.8.3 Inputs — JJCOMP, LSILS, ILSFLG
X, Y, H, RADTOD, DEGTOR

A.8.4 Outputs — EGS, ELOC, HSLOPE, RGS, RLOC, RKINK
XGS, XLOC, YLOC, SGS, SLOC, USLOPE, BSLOPE, HKINK,
CILS1, CILS2, CILS3, CILS4, XKINK

A.8.5 User options

LSILS	ILS on/off flag
0	off, glide slope and localiser indications set to zero
1	on
ILSFLG	select beam type
1	3^{o} straight beam
2	6^{o} straight beam
3	6^{o} changing to 3^{o} at height defined by HKINK

A.8.6 Description

A.8.6.1 If ILS is required (set by LSILS), calculates conventional angular ILS beam, with no beam noise, assuming *receiver in aircraft is at centre of gravity*.

A.8.6.2 If ILS is not required, glide slope and localiser indications are set to zero and no other calculations are performed.

A.8.6.3 Localiser model

Beam origin is at XLOC, YLOC. (X = 0 is at runway threshold, Y = 0 on runway centre line).

RLOC is range from transmitter.

Angular error ELOC is returned scaled to ±1.0, *ie* as a fraction of the defined beam width.

Beam width is ±SLOC degrees.

A.8.6.4 Glide slope model

Glide path origin is at XGS.

RGS is range from transmitter.

HSLOPE is height of beam centre line at range RGS.

Angular error EGS is returned scaled to ±1.0, *ie* as a fraction of the defined beam width.

Beam width is ±SGS degrees.

USLOPE is upper segment slope (if applicable) in degrees.

BSLOPE is lower segment slope in degrees.

For a single segment beam, USLOPE = BSLOPE.

If a 2-segment slope is used (selected by ILSFLG), the kink occurs at the defined height HKINK.

Range and position from kink (RKINK, XKINK) are derived.

A.8.7 Initialisation

A.8.7.1 During initialisation (when JJCOMP = 0) beam constants are set up according to the choice of beam defined by ILSFLG. Initialisation occurs regardless of the state of the ILS on/off flag.

A.8.8 Subroutines used

BOUND

Library mathematical routines.

A.8.9 Remarks

A.8.9.1 SILS is used by SINIT, if requested, to place the aircraft exactly on the beam as an initial condition.

A.8.9.2 If the user wishes to create his own ILS routine (or similar guidance) he could use the same name and ensure that it will interface correctly to SINIT, *ie* receive a value X and return a value HSLOPE.

A.9 Subroutine SINIT

A.9.1 Purpose Various initialisation calculations

A.9.2 Call CALL SINIT

A.9.3 Inputs

XIX, XIY, XIZ, XIZX
HIC, XIC, HSLOPE
DENR, SPSNDR, TEMPR, PRESSR
SPSL, RHOSL, TEMPSL, PRESSL
VWKTØ, PSIWD, SHRFAC
PSIDIC, GAMDIC, VKTIC, BETADIC, ALFADIC
SWREF, SPAN, W, XCG, ZCG, XCGREF, ZCGREF
KISA, ISHR

A.9.4 <u>Outputs</u>

CI1 - CI1Ø
X
ROOTSIG, SPSOND, RHO, TALPHA, PALPHA
CPSIW, SPSIW, VWNLØ, VWELØ, VWN, VWE, VWD
VKNIC, VKEIC, VKDIC, VN, VE, VD, VK, VT
THETADIC, PHIDIC
SB2, XMASS, RXMASS, DXCG, DZCG
PDIC, QDIC, RDIC
RADTOD, DEGTOR, FPSTKT, XK2FPS, G, RG

A.9.5 <u>Data to be supplied by user</u>

SWREF, SPAN, XCGREF, ZCGREF, ZCG, XIX, XIY, XIZ, XIZX

A.9.6 <u>User options</u>

KISA	flag to control variation of atmospheric properties with height
0	sea level conditions at all times
1	standard atmosphere
-1	fixed values appropriate to initial height (HIC)
ISHR	wind shear selection flag (see SWIND for full description)

A.9.7 <u>Description</u>

A.9.7.1 Calculates inertia coefficients, according to equations (20)

```
CIØ  = 1.0/(XIZ*XIX - XIZX*XIZX)
CI1  = XIZ*CIØ
CI2  = XIZX*CIØ
CI3  = XIZX*(XIX - XIY + XIZ)*CIØ
CI4  = (XIZ*(XIY - XIZ) - XIZX*XIZX)*CIØ
CI5  = 1.0/XIY
CI6  = XIZX*CI5
CI7  = (XIZ - XIX)*CI5
CI8  = XIX*CIØ
CI9  = (XIX*(XIX - XIY) + XIZX*XIZX)*CIØ
CI1Ø = -XIZX(XIX - XIY + XIZ)*CIØ
```

A.9.7.2 If HIC is set negative, calculates initial height to put aircraft exactly on ILS beam.

A.9.7.3 Sets up initial values of atmospheric properties. If KISA = 0, sets up sea level conditions. Otherwise uses SATMOS to derive properties appropriate to HIC.

A.9.7.4 Calculates local true wind components from datum wind speed $(V_{WKT\emptyset})$ and direction (ψ_W)

$$V_{WNL\emptyset} = -V_{WKT\emptyset} \cos \psi_W$$

$$V_{WEL\emptyset} = -V_{WKT\emptyset} \sin \psi_W$$

and allows for wind shear to give initial values for V_{WN}, V_{WE}.

A.9.7.5 If initial condition is not at rest derives velocity relative to ground, and then components for IC values, given wind speed and direction, desired initial airspeed and heading. (Note that VK in this routine is resultant velocity relative to ground, whereas VK obtained in SPATH is truly the ground speed, *ie* in the horizontal plane.)

An iteration technique is used to ensure that the initial track and heading are compatible with the specified wind. The initial heading will thus not reveal to the pilot anything about the applied wind.

A.9.7.6 If initial condition *is* at rest (defined by VKTIC < 1.0) then *ground* speed is set to zero.

A.9.7.7 An initial value for θ is calculated, assuming $\emptyset = 0$, from

$$\theta = \gamma + \alpha .$$

A.9.7.8 Aircraft related constants are calculated from data supplied by the user

```
SB2    = SWREF*SPAN*SPAN
XMASS  = W/G
RXMASS = 1.0/XMASS
DXCG   = XCG - XCGREF
DZCG   = ZCG - ZCGREF
```

A.9.7.9 Various system constants are given default values via DATA statements, as detailed below.

Default values

RADTOD	57.295780
DEGTOR	0.0174533
FPSTKT	0.5921053
XK2FPS	1.688889
G	32.174
RHOSL	0.0023769
SPSL	1116.45
RG	0.03108
TEMPSL	288.15
PRESSL	14.69597
DENR	1.0
SPSNDR	1.0
TEMPR	1.0
PRESSR	1.0
PDIC	0.0

QDIC	0.0
RDIC	0.0
PHIDIC	0.0
KISA	0
ISHR	1

A.9.7.10 A software switch is also set on (IHSHIO = 1) to enable the high speed hybrid input/output. It is communicated via labelled COMMON HSHIO.

A.9.8 Initialisation

Whole routine is for initialisation only.

A.9.9 Subroutines used

SATMOS, SILS, WSHEAR and library functions.

A.9.10 Remarks

Will not initialise correctly if initial ϕ is non-zero.

A.10 Subroutine SPATH

A.10.1 Purpose Calculates velocity over ground and flight path angles

A.10.2 Call CALL SPATH

A.10.3 Inputs VKN, VKE, VKD
RADTOD, FPSTKT

A.10.4 Outputs VK, VKKT,
PSIKR, PSIKD,
GAMMAR, GAMMAD

A.10.5 Description

A.10.5.1 Calculates velocity over ground (ground speed)

$$VK = (VKN^2 + VKE^2)^{\frac{1}{2}}$$

A.10.5.2 Converts VK to knots and stores.

A.10.5.3 Calculates climb angle

$$\gamma = \tan^{-1}(-VKD/VK)$$

in the range $-\pi/2$ to $+\pi/2$.

A.10.5.4 Calculates track

)08

$$\chi (= \psi_k) = \tan^{-1}(V_{KE}/V_{KN})$$

in the range 0 to 2π.

A.10.5.5 Converts γ and χ to degrees and stores.

A.10.6 Initialisation

None

A.10.7 Subroutines used

Library mathematical functions.

A.10.8 Remarks

A.10.8.1 If VKD = VK = 0.0 , $\gamma = 0.0$.

A.10.8.2 If VKE = VKN = 0.0 , $\chi = 0.0$.

A.10.8.3 For $V_{KE} = 0.0$, $\chi = 0$ for V_{KN} positive but $\chi = 180^{\circ}$ for V_{KN} negative.

A.11	Subroutine	STV
A.11.1	Purpose	Computes TV positions and rates, and controls belt logic.
A.11.2	Call	CALL STV
A.11.3	Inputs	VKN, VKE, VKD, VSHIPKT, XK2FPS X, Y, H, XIC XP, ZP, CTHETA, STHETA, CPSI, SPSI, PSIDT, THETDT JJCOMP, NTVB LFWD, LBACK, LYTVIC, LXTVIC, LCYCLE
A.11.4	Outputs	XTV, YTV, HTV, XIC, XIHX, XIHY XDOTM, XDTV, YDTV, HDTV SXPL, SXMIN, SXPLUS, SXMINUS, SYPL, SYMIN, HTVLIM, XDTMX IDSYNC

A.11.5 Data to be supplied by user

NTVB	TV belt flag
1	700:1 model
2	2000:1 model
3	5000:1 model

VSHIPKT ship velocity, knots.

A.11.6 Description

A.11.6.1 Initialisation.

A.11.6.2 Belt slew control.

If forward slew is selected, X rate is forced to maximum positive value.

If backward slew is selected, X rate is forced to maximum negative value.

In both cases, the aircraft's position keeps in step. Positional calculations can also be referred to a ship moving at constant velocity VSHIPKT.

A.11.6.3 Integration control for X and Y (belt logic).

(a) If Y reset is selected, Y position resets to YIC.

(b) If X reset is selected, X position resets to XIC as originally defined in ICFILE.

(c) If X reset is not selected, additional logic is invoked to control the belt (x):

- if X is not near the belt join (defined by SXMIN, SXPL - see belt diagram), the program continues to the position module.
- If X is near the join but 'LCYCLE' is not set, the program continues to the position module.
- If 'LCYCLE' is not set, the program continues to the position module.
- If 'LCYCLE' is set, X is forced to cycle between SXMINUS and SXPLUS. IDSYNC is set to 1 to uncouple the belt position feedback.
- If 'LCYCLE' is set, but the demanded X position is less than the negative cycle limit SXMINUS (as can happen if X had been previously reset), the X integrator is reset and XIC is set to SXMINUS. This will force the belt to start to move.
- If X is between SXMINUS and SXPLUS, the program continues to the position module.
- When X reaches SXPLUS, it resets to SXMINUS, before going on to the position module.

A.11.6.4 Position module - TV positions and rates in three axes (equations (22) to (25)).

(a) Calculates pilot position, and rate of change of his position, allowing for his location away from the aircraft's centre of gravity.

(b) If TV belt is trying to position before minimum X possible on the belt (SXMIN), or if X reset is demanded, then $\dot{X}_{TV}$ is set to zero to stop hunting. This is not done if LCYCLE is set.

(c) If H is above the ceiling for the model, $\dot{h}_{TV}$ is set to zero to stop bouncing.

(d) Limits are finally applied to x_{TV}, y_{TV} and h_{TV} to keep within model confines. These limits do not affect computed aircraft position and navigation.

A.11.7 Initialisation

A.11.7.1 During initialisation, the set of TV constants appropriate to the particular belt are selected from the array TVCON (8,3). These constants define the limit of movement as follows:

SXPL	TVCON (1, NTVB)	Maximum value of x_{TV} allowed, also determines when 'near belt join'.
SXMIN	TVCON (2, NTVB)	Minimum value of x_{TV} allowed, also determines when 'near belt join'.
SXPLUS	TVCON (3, NTVB)	x cycles between these two values if
SXMINUS	TVCON (4, NTVB)	'cycle' mode is selected
SYPL	TVCON (5, NTVB)	Maximum value of y_{TV} allowed
SYMIN	TVCON (6, NTVB)	Minimum value of y_{TV} allowed
HTVLIM	TVCON (7, NTVB)	Ceiling for TV height
XDTMX	TVCON (8, NTVB)	Belt velocity when slewing.

The x-related quantities are illustrated in Fig A4.

STV holds three values of each parameter in TVCON, the appropriate one being selected according to the TV belt in use. Because of this, on-line permanent changes to SXPL etc can only be achieved by changing the associated element of TVCON.

A.11.7.2 The initial value of X obtained from the IC file is stored in a temporary location (XICF) so that a demand to reset X will reset X to this value regardless of whether cycle has previously been selected.

A.11.8 Subroutines used

BOUND

A.11.9 Remarks

A.11.9.1 The array of TV constants is accessible to PV100 and so can be overwritten if new values are found to be more appropriate to the hardware. Alterations can also be made via the semi-permanent changes file (see SYSCOM).

A.11.9.2 The logical controls are set via special buttons etc on the control desk and so are hardware dependent.

LCYCLE	desk switch 5
LXTVIC	desk switch 6
LYTVIC	desk switch 7
LFWD	slew toggle patched to AD4 sense line DGS03 (line 4)
LBACK	slew toggle patched to AD4 sense line DGS04 (line 5)
IDSYNC	software flag controls AD4 control line DGC17 (line 16)

A.11.9.3 To start the aircraft at some position outside the limits of the TV model, set up XIC, YIC to suit. When the aircraft position comes within the confines of the belt, it will begin to move with the aircraft. X and Y positions can be reset using their appropriate buttons without having to reset the whole computation.

A.11.9.4 If, after a reset to some distant location, immediate movement of the belt is required, put 'X reset' off and 'cycle' on.

A.11.9.5 No provision has been made for large bank angles (>1 rev), nor for cloud or visibility control.

A.11.9.6 Successful use of this routine to drive the TV model is dependent on analogue computer circuitry incorporating position feedbacks.

A.11.9.7 Do not initialise with 'CYCLE' on if, subsequently in the run, you may want to reset X outside the confines of the belt. This is because initialisation occurs twice and so XIC is overwritten.

A.12 Subroutine SVELOC1

A.12.1 Purpose Calculates body axes velocity components

A.12.2 Call CALL SVELOC1

A.12.3 Inputs VKN, VKE, VKD
VWN, VWE, VWD
$S_{11} - S_{33}$

A.12.4 Outputs UB, VB, WB
VN, VE, VD

A.12.5 Description

A.12.5.1 Given the components of velocity relative to ground (VKN, VKE, VKD) and the components of total wind speed, including turbulence (VWN, VWE, VWD),
008 calculates components of velocity relative to the air in earth axes (VN, VE, VD), then resolves to body axes.

A.12.6 Initialisation

None

A.12.7 Subroutines used

None

A.12.8 Remarks

A.12.8.1 Assumes wind velocity component is positive in the *same* direction as the component relative to the air. Thus, for a head wind and for the aircraft on a northerly heading ($\psi = 0$), VWN is negative.

A.12.8.2 VN, VE, VD are initially calculated in SINIT.

A.13	Subroutine	SVELOC2
A.13.1	Purpose	Calculates total airspeed, equivalent airspeed, Mach number etc using appropriate atmospheric properties.
A.13.2	Call	CALL SVELOC2
A.13.3	Inputs	UB, VB, WB H SWREF, CREF, SPAN, SB2 KISA FPSTKT, RHOSL, SPSL, TEMPSL, PRESSL
A.13.4	Outputs	VT, VTKT, VEAS, VEASKT, XMACH QDYN, HLFROV, QDYNS, QSCREF, QSSPAN, QSB2IV ROOTSIG, RHO, SPSOND, DENR, SPSNDR, TEMPR, PRESSR TRATIO, ARATIO, DRATIO, PRATIO, TALPHA, PALPHA

A.13.5 Data to be supplied by user

Note that certain geometric parameters for the aircraft must be supplied by the user.

SWREF	reference wing area
CREF	reference chord
SPAN	reference wing span

SB2 is a combination (SWREF*SPAN*SPAN) calculated in SINIT.

A.13.6 User options

The flag KISA selects the way in which atmospheric properties vary. It must only be changed in INITIAL, before SINIT is executed.

KISA = 1	standard atmosphere
= 0	fixed values appropriate to sea level
= -1	fixed values appropriate to initial height HIC

008

A.13.7 Description

A.13.7.1 Calculates total airspeed, converts to knots and stores.

$$V_T = \left(u_B^2 + v_B^2 + w_B^2\right)^{\frac{1}{2}}$$

A.13.7.2 If the flag KISA is set to 1, obtains, for the current height, new properties (density ratio, speed of sound ratio, temperature ratio and pressure ratio) of a standard atmosphere, using the routine SATMOS.

If KISA is not set to 1, atmospheric properties are kept constant, as set up by SINIT.

A.13.7.3 Calculates properties of the atmosphere

air density	$\rho = \rho_{SL}\sigma$
speed of sound	$a = a_{SL}a_r$
ambient temperature	$T_\alpha = T_{SL}T_r$
ambient pressure	$P_\alpha = P_{SL}P_r$

A.13.7.4 Calculates equivalent airspeed, converts to knots and stores.

$$V_{EAS} = V_T\sigma^{\frac{1}{2}} .$$

A.13.7.5 Calculates dynamic pressure and related terms

$$\text{QDYN} = \tfrac{1}{2}\rho V_T^2$$

$$\text{QDYNS} = \tfrac{1}{2}\rho V_T^2 S_W$$

$$\text{QSCREF} = \tfrac{1}{2}\rho V_T^2 S_W c_{ref}$$

$$\text{QSSPAN} = \tfrac{1}{2}\rho V_T^2 S_W b$$

$$\text{QSB21V} = \tfrac{1}{2}\rho V_T (S_W b^2)$$

A.13.7.6 Calculates Mach number

$$M = V_T/a$$

A.13.7.7 Calculates compressible adiabetic flow relationships for Mach number <1.0. For initialisation purposes, these ratios are given datum values of 1.0.

$$T_{ratio} = (1 + 0.2M^2)^{-1}$$

$$a_{ratio} = (T_{ratio})^{\frac{1}{2}}$$

$$\rho_{ratio} = (a_{ratio})^5$$

$$P_{ratio} = a_{ratio}\ T_{ratio}$$

A.13.8 Initialisation

In addition to those items mentioned above, datum values of atmospheric properties are provided in SINIT (qv).

A.13.9 Subroutines used

SATMOS

A.13.10 Remarks

Subroutine ATMOS, an alternative to SATMOS, is available to calculate atmospheric properties for hot days. It is not included as an option, however, as it is very much larger than SATMOS, the subroutine for 'standard' days.

A.14	Subroutine	SWIND
A.14.1	Purpose	Controls turbulence, wind and wind shear
A.14.2	Call	CALL SWIND
A.14.3	Inputs	H, HIC, VKTIC, XK2FPS, FRAMET1, SFRACG, SRDECAY VWNLØ, VWELØ, VWDLØ, CPSIW, SPSIW USIG, VSIG, WSIG ISHR, JJCOMP, NG, LTURB, LSEED
A.14.4	Outputs	UTURB, VTURB, WTURB VWN, VWE, VWD VWNL, VWEL, VWDL SHRFAC

A.14.5 User options

LTURB	Turbulence switch
0	no turbulence calculated or output
1	turbulence calculated and output
ISHR	wind shear selection
1 (default)	no shear applied. Returns immediately with SHRFAC = 1.0. Time overhead is minimal.
2	ARB mean shear profile (logarithmic)
3	linear shear up to 333 ft, thereafter constant
LSEED	random number seed selector
0 (default)	seeds in GUSTS routine obtained from time of day
1	constant seeds enable a repeatable turbulence sequence to be obtained
USIG, VSIG, WSIG	rms settings for each component (never exactly achieved - it is a random process!). Can be changed in flight. 008
VWKTØ	datum wind speed, true (knots)

PSIWD — datum wind direction (degrees). PSIWD = 0.0 is a wind from the north. VWKTØ and PSIWD are set as initial conditions and are used by SINIT (qv) to calculate VWNLØ, CPSIW etc.

For the remaining parameters (except for VWDL) changes are only effective if applied during initialisation.

VWDL — wind component in 'down' sense (default = 0.0)

SRDECAY — decay parameter, should not be changed (default = 0.7)

SFRACG — intermittency parameter, in range 0.0 to 0.99 (default = 0.0). Should never be set equal to 1.0. For minimum intermittency, set SFRACG = 0.0. A value of 0.7 is probably as extreme as will ever be needed. Refer to Ref 12 for discussion.

NG — Controls the three gust ramp lengths used in the turbulence generation routine GUSTS[12], via

$$\text{NGUSTS} = \text{NG}^* \frac{\text{Initial speed (ft/s)}}{200} .$$

NGUSTS, an important parameter, required by the GUSTS routine, defines the number of gusts per second. Thus, if the initialisation speed is 200 ft/s (118 kn), NGUSTS = NG = 4 (the default value for NG) and so the shortest gust gradient distance is 200/4 = 50 ft and the longest, determined in the GUSTS subroutine, is 16 times the shortest, or 800 ft. Initialisation at speeds much below 100 kn may require NG to be increased. If the initial speed is zero, NGUSTS takes a minimum value of 1. A further limitation is that NGUSTS must be less than the repetition rate of the simulation, *eg* for a frame time of 50 ms (20 solutions/s) and NG = 4, the *maximum* initial speed must not exceed 1000 ft/s (592 kn). This constraint is checked and imposed in SWIND.

A.14.6 Description

(a) Derives shear factor (SHRFAC) according to height, using WSHEAR subroutine.

(b) Calculates local wind velocity components VWNL, VWEL at height, including shear effects.

VWNL = VWNLØ * SHRFAC
VWEL = VWELØ * SHRFAC
VWNLØ, VWELØ are calculated in SINIT.

(c) If turbulence is required (set by LTURB), calculates three raw components of turbulence (using GUSTS subroutine). The gust velocities obtained are scaled by nominal rms values for each
008 component (USIG, VSIG, WSIG), set by the user, to give turbulence components UTURB, VTURB, WTURB.

(d) If turbulence is not required, turbulence components are set to zero.

(e) The calculated turbulence components are added to the mean wind components to provide three components of total wind (VWN, VWE, VWD) in earth axes.

VWN = VWNL - (UTURB * CPSIW - VTURB * SPSIW)
VWE = VWEL - (VTURB * CPSIW + UTURB * SPSIW)
VWD = VWDL - WTURB

(f) A flow chart is shown in Fig A9.

A.14.7 Initialisation

During initialisation, the gust generation routine GUSTS is called regardless of whether the turbulence switch LTURB is set on or off. This is necessary to perform the required preliminary calculations. However, the turbulence components returned are set to zero. GUSTS uses its own flag (IFLAG) to control initialisation, but this is tied to the system flag JJCOMP in SWIND.

A.14.8 Subroutines used

WSHEAR, GUSTS, FDC, RANDU, VSTEP, SEEDVAL
Library mathematical functions.

A.14.9 Remarks

A.14.9.1 The calculated turbulence components are added to the mean wind components, which only later become resolved into body axes. UTURB is *along* the datum wind, a positive gust increasing the wind (crudely, increasing the airspeed); VTURB is across the wind (a positive gust, crudely, increasing the sideslip); WTURB is vertical, positive up (to increase incidence).

A.14.9.2 Wind shear and attenuation of vertical turbulence with height work from altitude rather than height above local terrain.

A.14.9.3 Details of the gust generation process are contained in Ref 12.

A.14.9.4 Wind shear is only in magnitude, not in direction.

A.14.9.5 The turbulence control, LTURB, is set by a desk switch (currently number 8).

A.15 Subroutine SYSCOM

A.15.1 Purpose Sets up communication with system variables.

A.15.2 Call CALL SYSCOM

A.15.3 Inputs As set up in 'changes' file.

A.15.4 Outputs Nil

A.15.5 Description

A.15.5.1 Includes in the labelled common area SYSTEM *all* the names of system variables so that they are set up in the Namelist table.

A.15.5.2 Calls PV200 in the defined way to set up the Namelist table address. The defined way is

```
      DIMENSION NAME (2)
      DATA NAME/'SYSCOM'/
      NAMELIST
      CALL PV200(NAME)
      GO TO 3
2     INPUT 101
3     CONTINUE
```

A.15.5.3 Provides for 'semi-permanent' changes to be read from a file, using the self-identified input facility

```
      DATA ISYS/50/
      REWIND ISYS
      INPUT (ISYS)
      REWIND ISYS
```

A.15.6 Initialisation

This routine is only called in the INITIAL region, so it is all initialisation.

A.15.7 Subroutines used

PV200 plus library routines. Note that to test SYSCOM in the background, a dummy PV200 should be supplied.

A.15.8 Remarks

A.15.8.1 Unit 50 used for inputting semi-permanent changes should be assigned at model load time to a file in the user's area, *eg*

:ASSIGN (F:50, D3, SYSCHNG)

A.15.8.2 Prior to running the aircraft model program, the 'changes' file should be filled with the desired changes in the form

```
X1    = 23.0
SWREF = 560.0
*
```

The concluding asterisk (*) is essential. If no changes are required the file *must contain* an asterisk otherwise a run-time error occurs.

A.15.8.3 The positioning of SYSCOM means that IC values cannot be overwritten since they are read after SYSCOM is executed. However, changes introduced through SYSCOM can be altered on-line since retained changes are applied (by PV300) after SYSCOM.

A.16 <u>Subroutine</u> DSCRT1/DSCRT2

A.16.1 <u>Purpose</u> To control execution of routines for discrete input/output.

A.16.2 <u>Call</u> CALL DSCRT1 (CALL DSCRT2)

A.16.3 <u>Inputs</u> IAD4SL, IAD4CL, IDSCFL, IPCOFL

A.16.4 <u>Outputs</u> By calling other routines, the arrays ISLR(16), ICLR(16), IP(32), IDS(32), ICO(32) are filled.

A.16.5 <u>Data to be supplied by user</u>

Four flags must be set to control which of the basic routines, if any, is executed in which real time loop.

IAD4SL	Execution control flag for AD4 sense lines (READSLR)
IAD4CL	Execution control flag for AD4 control lines (SETCLR)
IDSCFL	Execution control flag for discretes in to the Sigma (READSCR)
IPCOFL	Execution control flag for discretes out from Sigma (SETDSCR)

Each flag can take the values:

0	Do not execute
1	Execute in DSCRT1, *ie* in the fastest loop
2	Execute in DSCRT2, *ie* in the second loop.

Default values (set in DSCRT1) and typical values are

Flag	Routine	Default	Typical	(possible)
IAD4SL	READSLR	1	1	(2)
IAD4CL	SETCLR	1	1	(2)
IDSCFL	READSCR	1	2	
IPCOFL	SETDSCR	0	2	(1)

A.16.6 <u>Description</u>

Calls appropriate routines, according to flag settings. Routines which are called by DSCRT1 are not then called by DSCRT2, and *vice versa*. 008

A.16.7 Initialisation

Nil

A.16.8 Subroutines used

READSLR, SETCLR, READSCR, SETDSCR

A.16.9 Remarks

Execution timings are

SETCLR	0.16 ms
READSLR	0.22 ms
SETDSCR	0.34 ms
READSCR	0.87 ms
Total	1.59 ms

Hence a millisecond or more could be saved in the main loop by executing READSCR and SETDSCR in the second loop.

A.17 Function subroutine ISDSCR

A.17.1 Purpose — To read a single specified Sigma 8 sense line

A.17.2 Call — I = ISDSCR (ISET, LINE)
ISDSCR is returned with a value 0 or 1, depending on whether the specified line LINE is off or on.

A.17.3 Description

This function tests the status of an individual sense line, either from the logic patch panel in rack N(ISET = 0) or from the control desk switches (ISET = 1).

A.17.4 Initialisation

Nil

A.17.5 Subroutines used

READSCR

A.17.6 Remarks

This function is intended for use during the INITIAL region only, when time is not critical. It is not a fast routine.

Appendix B

LISTING OF SESAME ROUTINES

```
      SUBROUTINE SACCBOD
C
C
C               ************************************************
C               *
C               *   WRITTEN BY : A.J.SPENCER                     *
C               *   DATE :  24.02.78                             *
C               *   DESCRIPTION : SACCBOD CALCULATES ACCELERATIONS *
C               *     OF THE AIRCRAFT C.G. AND OF THE PILOT , IN *
C               *     THE BODY AXES , FOR USE BY MOTION , AUTOSTAB *
C               *     ETC..                                      *
C               *                                                *
C               ************************************************
      COMMON/SYSTEM/A(1000),C(400)
C
C     VERSION IDENTIFIER
C
      DATA KSACCBOD/240278/
C     12-8-76      SYSTEM COMMON ENLARGED
C  24.2.78 XP,ZP REPLACED BY XPCG,ZPCG
      EQUIVALENCE(KSACCBOD,C(255))
C
C     *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(AXCG           ,A(241)),(AYCG           ,A(242)),(AZCG           ,A(243)),
     1           (AZCG1          ,A(244)),(AXACG          ,A(245)),(AYACG          ,A(246)),
     2           (AZACG          ,A(247)),(AXP            ,A(248)),(AYP            ,A(249)),
     3           (AZP            ,A(250)),(AZAP           ,A(251)),(AY1            ,A(252)),
     4           (AZ2            ,A(253)),(AX3            ,A(254)),(AY4            ,A(255)),
     5           (AZ5            ,A(256))
C
C     *** VARIABLE INPUTS ***
C
      EQUIVALENCE(X1             ,A(257)),(Y1             ,A(258)),(Z1             ,A(259)),
     1           (X2             ,A(260)),(Y2             ,A(261)),(Z2             ,A(262)),
     2           (X3             ,A(263)),(Y3             ,A(264)),(Z3             ,A(265)),
     3           (X4             ,A(266)),(Y4             ,A(267)),(Z4             ,A(268)),
     4           (X5             ,A(269)),(Y5             ,A(270)),(Z5             ,A(271)),
     5           (S13            ,A( 19)),(S23            ,A( 22)),(S33            ,A( 25)),
     6           (P              ,A( 26)),(Q              ,A( 27)),(R              ,A( 28)),
     7           (PDOT           ,A( 45)),(QDOT           ,A( 46)),(RDOT           ,A( 47)),
     8           (XPCG           ,A(280)),(ZPCG           ,A(281)),(FTX            ,A(134)),
     9           (FTY            ,A(135)),(FTZ            ,A(136)),(W              ,A(198))   240278
C
C     **** CONSTANTS ****
C
      EQUIVALENCE(RG             ,A(146))
C
C  EVALUATING COMMON SUB-EXPRESSIONS
C
      P2 = P*P
      Q2 = Q*Q
      R2 = R*R
      PQ = P*Q
      PR = P*R
      QR = Q*R
C  THESE INCLUDE DIVISION BY 'G'
      Q2R2   = (Q2 + R2)*RG
      PQRDTB = (PQ-RDOT)*RG
      PRQDT  = (PR+QDOT)*RG
      PQRDT  = (PQ+RDOT)*RG
      P2R2   = (P2+R2)*RG
      QRPDTB = (QR-PDOT)*RG
      PRQDTB = (PR-QDOT)*RG
      QRPDT  = (QR+PDOT)*RG
      P2Q2   = (P2+Q2)*RG

      RW = 1.0/W
C
C  CALCULATING ACCELERATIONS IN 'G'UNITS
C
C  SPECIFIC ACCNS AT AIRCRAFT C.G.
C
      AXCG = FTX*RW
```

```
      AYCG = FTY*HW
      AZCG = FTZ*HW
C
C  ABSOLUTE ACCNS AT AIRCRAFT C.G.
C
      AXACG = AXCG+S13
      AYACG = AYCG+S23
      AZACG = AZCG+S33
C
C  SPECIFIC ACCNS AT PILOT
C
      AXP = AXCG-XPCG*Q2R2 + ZPCG*PRQDT                                 2*0278
      AYP = AYCG+XPCG*PQRDT + ZPCG*QRPDTB                               2*0278
      AZP = AZCG+XPCG*PRQDTA - ZPCG*P2Q2                                2*0278
C
C  ABSOLUTE (OR 'MANUEUVRE G') ACCNS AT PILOT
C
      AZAP = AZP+S33
C
C  ACCNS FOR RECORDERS RELATIVE TO 1.0 G
C
      AZCG1 = AZCG+1.0
C
C  ACCNS FOR SLIP BALL (LOCATED AT X1,Y1,Z1)
C
      AY1 = AYCG+X1*PQRDT-Y1*P2R2+Z1*QRPDTB
C
C  ACCNS FOR 'G' METER (AT X2,Y2,Z2)
C
      AZ2 = AZCG+X2*PRQDTB+Y2*QRPDT-Z2*P2Q2
C
C ACCNS FOR THREE INDEPENDANT ACCELEROMETERS (FOR USE BY AUTOSTAB)
C             X-ACCELEROMETER AT X3,Y3,Z3
C             Y-ACCELEROMETER AT X4,Y4,Z4
C             Z-ACCELEROMETER AT X5,Y5,Z5
C
      AX3 = AXCG-X3*Q2R2+Y3*PQRDTB+Z3*PRQDT
      AY4 = AYCG+X4*PQRDT-Y4*P2R2+Z4*QRPDTB
      AZ5 = AZCG+X5*PRQDTB+Y5*QRPDT-Z5*P2Q2
      RETURN
      END
!ASSIGN (M:SI,D3,SACLS)

!FORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE SACCLIN
C
C        ***************************************************
C        *                                                 *
C        *    WRITTEN BY: D.OLDFIELD                       *
C        *    DATE:12-8-76                                 *
C        *    DESCRIPTION: THIS SUBROUTINE CALCULATES THE TOTAL *
C        *        FORCE COMPONENTS TO THE EARTH AXES GIVEN THE  *
C        *        COMPONENTS TO THE BODY AXES AND THE DIRECTION *
C        *        COSINES. THE ACCELERATION IS ALSO CALCULATED  *
C        *        FROM THIS INFORMATION AND THE AIRCRAFT MASS   *
C        *                                                 *
C        ***************************************************
C
C     12-8-76     SYSTEM COMMON ENLARGED
      COMMON/SYSTEM/A(1000),L(400)
C
C   VERSION IDENTIFIER
C
      DATA KSACCLIN/120876/
      EQUIVALENCE(KSACCLIN,L(256))
C
C              *** VARIABLE INPUTS ***
C
      EQUIVALENCE(S11     ,A( 17)),(S12     ,A( 18)),(S13     ,A( 19)),
     1           (S21     ,A( 20)),(S22     ,A( 21)),(S23     ,A( 22)),
     2           (S31     ,A( 23)),(S32     ,A( 24)),(S33     ,A( 25)),
     3           (FTX     ,A(134)),(FTY     ,A(135)),(FTZ     ,A(136)),
     4           (RXMASS  ,A(  1))
C
C              *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(FTN     ,A(137)),(FTE     ,A(138)),(FTD     ,A(139)),
     1           (VKNDOT  ,A(238)),(VKEDOT  ,A(239)),(VKDDOT  ,A(240))
C
C              *** CONSTANTS ***
C
      EQUIVALENCE(G       ,A(145))
C
C         CALCULATE TOTAL FORCES TO EARTH AXES
      FTN = S11*FTX+S21*FTY+S31*FTZ
      FTE = S12*FTX+S22*FTY+S32*FTZ
      FTD = S13*FTX+S23*FTY+S33*FTZ
C         CALCULATE ACCELERATION
      VKNDOT= FTN*RXMASS
      VKEDOT= FTE*RXMASS
      VKDDOT= FTD*RXMASS+G
      RETURN
      END.
IASSIGN (MISI,D3,SACRS)

IFORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE SACCROT
C
C         ***************************************************
C         *                                                 *
C         *    WRITTEN BY : D.BLUFIELD                      *
C         *    DATE : 12-8-76                               *
C         *    DESCRIPTION : THIS SUBROUTINE CALCULATES     *
C         *      THE ANGULAR ACCELERATIONS IN THE BODY      *
C         *      AXES GIVEN THE TOTAL MOMENTS,ANGULAR       *
C         *      VELOCITY COMPONENTS IN THE BODY AXES AND   *
C         *      THE INERTIA COEFFICIENTS.                  *
C         *                                                 *
C         ***************************************************
C
C     12-8-76     SYSTEM COMMON ENLARGED
      COMMON/SYSTEM/A(1000),L(400)
C
C   VERSION IDENTIFIER
C
      DATA KSACCROT/120876/
      EQUIVALENCE(KSACCROT,L(257))
C
C             *** VARIABLE INPUTS ***
C
      EQUIVALENCE(XLLTOT  ,A(131)),(XMMTOT  ,A(132)),(XNNTOT  ,A(133)),
     1           (P       ,A( 26)),(Q       ,A( 27)),(R       ,A( 28)),
     2           (CI1     ,A( 35)),(CI2     ,A( 36)),(CI3     ,A( 37)),
     3           (CI4     ,A( 38)),(CI5     ,A( 39)),(CI6     ,A( 40)),
     4           (CI7     ,A( 41)),(CI8     ,A( 42)),(CI9     ,A( 43)),
     5           (CI10    ,A( 44))
C
C             *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(PDOT    ,A( 45)),(QDOT    ,A( 46)),(RDOT    ,A( 47))
C
C         CALCULATE ANGULAR ACCELERATIONS
      PDOT = CI1*XLLTOT+CI2*XNNTOT+(CI3*P+CI4*R)*Q
      QDOT = CI5*XMMTOT+CI6*(R*R-P*P)+CI7*R*P
      RDOT = CI8*XNNTOT+CI2*XLLTOT+(CI9*P+CI10*R)*Q
C
      RETURN
      END
!ASSIGN (M:SI,D3,SALFS)

!FORTRAN LS,NB
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE SALFBLT(DT)
C
C         ***********************************************
C         *                                             *
C         *    WRITTEN BY : D.OLDFIELD                  *
C         *    DATE : 21-3-78                           *
C         *    DESCRIPTION : THIS SUBROUTINE CALCULATES *
C         *      THE ANGLE OF ATTACK AND SIDE SLIP. THE *
C         *      SINE AND COSINE OF THE ANGLES ARE ALSO *
C         *      CALCULATED. THE DERIVATIVES OF THE ANGLES *
C         *      ARE DERIVED FROM THE ASSUMPTION OF A   *
C         *      QUADRATIC FIT TO THE DATA USING THE    *
C         *      TIME INTERVAL IS PASSED AS A PARAMETER *
C         *                                             *
C         ***********************************************
C
C      12-8-78     SYSTEM COMMON ENLARGED
C      21.3.79     TEMP VARIABLE NAME CHANGED
C       9.5.79     EVALUATION OF DERIVATIVES CORRECTED
      COMMON/SYSTEM/A(1000),L(1000)
C
C    VERSION IDENTIFIER
C
      DATA KSALFBLT/090679/
      EQUIVALENCE(KSALFBLT,L(258))
      DATA I/0/
C
C             *** VARIABLE INPUTS ***
C
      EQUIVALENCE(UB     ,A( 89)),(VB      ,A( 90)),(WB      ,A( 91))
      EQUIVALENCE(NCPASS ,L(  2)),(NIPASS  ,L(  3))                        090679
C
C             *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(ALFAR  ,A(111)),(ALFAD   ,A(112)),(ALFADOT ,A(120)),
     1           (SALFA  ,A(116)),(CALFA   ,A(117)),(BETAR   ,A(113)),
     2           (BETAD  ,A(114)),(BETADOT ,A(121)),(SBETA   ,A(118)),
     3           (CBETA  ,A(119))
C
C             *** CONSTANTS ***
C
      EQUIVALENCE(RADTOD ,A(145))
C
C          CALCULATE ANGLE OF ATTACK,IN RANGE +/- 180 DEG.
      ALFAR = 0.0
      IF(WB.EQ.0.0.AND.UB.EQ.0.0) GO TO 7
      ALFAR = ATAN2(WB,UB)
    7 ALFAD = ALFAR*RADTOD
      SALFA = SIN(ALFAR)
      CALFA = COS(ALFAR)
C          CALCULATE ANGLE OF SIDESLIP,IN RANGE +/- 90 DEG.
      V = SQRT(UB*UB+WB*WB)
      BETAR = 0.0
      IF(V.EQ.0.0.AND.VB.EQ.0.0) GO TO 8
      BETAR = ATAN2(VB,V)
    8 BETAD = BETAR*RADTOD
      SBETA = SIN(BETAR)
      CBETA = COS(BETAR)
C          SET UP PREVIOUS VALUES FOR FIRST ENTRY
      IF(I.GT.0) GO TO 10
      ALFARN1 = ALFAR
      ALFARN2 = ALFAR
      BETARN1 = BETAR
      BETARN2 = BETAR
      I = 1
   10 CONTINUE
C          CALCULATE DERIVATIVE OF ALFA AND BETA,THEN SET UP PREVIOUS VALUES
C          FOR NEXT ENTRY.
C
C    EVALUATE DERIVATIVES ONLY DURING LAST PASS OF INTEGRATION STEP
C    (AND DURING INITIALISATION)
C
      IF(NCPASS.LT.NIPASS-1)RETURN                                         090679
C
      RT = 0.5/DT                                                          210379
      ALFADOT = (ALFARN2-4.0*ALFARN1+3.0*ALFAR)*RT                         210379
      BETADOT = (BETARN2-4.0*BETARN1+3.0*BETAR)*RT                         210379
      ALFARN2 = ALFARN1
      ALFARN1 = ALFAR
      BETARN2 = BETARN1
      BETARN1 = BETAR
      RETURN
      END
```

ICC
1F1N

```
      SUBROUTINE SCOUNT
C
C         **************************************************
C         *                                                *
C         *    DESCRIPTION : THIS ROUTINE SETS PASS COUNT FOR    *
C         *      DERIVATIVE SECTION.                       *
C         *    WRITTEN BY : B.N.TOMLINSON                  *
C         *    DATE :12-8-76                               *
C         * MODIFIED: 31-5-77 TO INCLUDE DLRUN             *
C         *                                                *
C         **************************************************
C
C
C   VERSION IDENTIFIER
C
      DATA KSCOUNT/310577/                                          310577
      EQUIVALENCE(KSCOUNT ,L(259))
C     12-8-76     SYSTEM COMMON ENLARGED
      COMMON/SYSTEM/A(1000),L(400)
C  PICK UP JCOMP (SET IN SWAIT)
      COMMON/CONTROL/JCOMP
C  JCOMP = 1 AFTER COMPUTE PRESSED
C
C             *** VARIABLE INPUTS ***
C
C
C             *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE (NRUN    ,L(169))                                 310577
      EQUIVALENCE(NCPASS  ,L( 2)),(JJCOMP  ,L(  8))
C     NCPASS IS COUNT OF NUMBER OF PASSES THROUGH DERIVATIVE SECTION
C     EXECUTED SO FAR IN ONE INTEGRATION STEP. SET TO ZERO IN THIS
C     ROUTINE PRIOR TO DOING INTEGRATION.
      NCPASS = 0
C  SET JJCOMP,IN SYSTEM COMMON,TO ENABLE STATUS TO BE
C      USED,E.G. IN SILS,STV
      JJCOMP = JCOMP
C  PICK UP RUN NUMBER FROM MAILBOX
      CALL DLRUN (NRUN)                                             310577
      RETURN
      END
IASSIGN (MISI,D3,SDCSS)

IFORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
            SUBROUTINE SDCOS
      C
      C
      C     ****************************************************
      C     *                                                  *
      C     *   WRITTEN BY : D.OLDFIELD                        *
      C     *   DATE : 12-8-76                                 *
      C     *   DESCRIPTION : THIS SUBROUTINE  CALCULATES      *
      C     *      THE DIRECTION COSINES,SINES AND DEGREE      *
      C     *      VALUE OF THREE INPUT ANGLES IN RADIANS.     *
      C     *      SEC THETA AND TAN THETA ARE ALSO            *
      C     *      CALCULATED.                                 *
      C     *                                                  *
      C     ****************************************************
      C     12-8-76      SYSTEM COMMON ENLARGED
            COMMON/SYSTEM/A(1000),L(400)
      C
      C   VERSION IDENTIFIER
      C
            DATA KSDCOS  /120876/
            EQUIVALENCE(KSDCOS   ,L(260))
      C
      C
      C             *** VARIABLE INPUTS ***
            EQUIVALENCE(PSIR        ,A(  3)),(THETAR  ,A(  5)),(PHIR      ,A(  7))
      C
      C
      C             *** VARIABLE OUTPUTS ***
            EQUIVALENCE(PSID
           1           (SPSI     ,A(  4)),(THETAD  ,A(  6)),(PHID
           2           (CTHETA   ,A(  9)),(CPSI    ,A( 10)),(STHETA    ,A(  8)),
           3           (SECTHT   ,A( 12)),(SPHI    ,A( 13)),(CPHI      ,A( 11)),
           4           (S12      ,A( 15)),(TANTHT  ,A( 16)),(S11       ,A( 14)),
           5           (S22      ,A( 18)),(S13     ,A( 19)),(S21       ,A( 17)),
           6           (S32      ,A( 21)),(S23     ,A( 22)),(S31       ,A( 20)),
      C                          ,A( 24)),(S33     ,A( 25))            ,A( 23)),
      C
      C             *** CONSTANTS ***
      C
      C
            EQUIVALENCE(RADTOD   ,A(141))
      C
      C
                 CALCULATE SINE,COSINE AND TANGENTS
            STHETA = SIN(THETAR)
            CTHETA = COS(THETAR)
            SPSI   = SIN(PSIR)
            CPSI   = COS(PSIR)
            SPHI   = SIN(PHIR)
            CPHI   = COS(PHIR)
      C
                 CALCULATE DIRECTION COSINES
            S11 = CTHETA*CPSI
            S12 = CTHETA*SPSI
            S13 = -STHETA
            S21 = SPHI*STHETA*CPSI-CPHI*SPSI
            S22 = SPHI*STHETA*SPSI+CPHI*CPSI
            S23 = SPHI*CTHETA
            S31 = CPHI*STHETA*CPSI+SPHI*SPSI
            S32 = CPHI*STHETA*SPSI-SPHI*CPSI
            S33 = CPHI*CTHETA
      C
                 CALCULATE SEC THETA AND TAN THETA
            SECTHT = 1.0/CTHETA
            TANTHT = STHETA*SECTHT
      C
                 CALCULATE ANGLES IN DEGREES
            PSID   = PSIR*RADTOD
            THETAD = THETAR*RADTOD
            PHID   = PHIR*RADTOD
            RETURN
            END
ìASSIGN (M:S1,D3,SEULS)

ìFORTRAN LS,NB
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE SEULER
C
C         ****************************************************
C         *                                                  *
C         *   WRITTEN BY : D.OLDFIELD                        *
C         *   DATE : 12-8-76                                 *
C         *   DESCRIPTION : THIS SUBROUTINE CALCULATES       *
C         *      THE ALTITUDE RATES GIVEN THE COMPONENTS     *
C         *        OF THE ANGULAR VELOCITY IN THE BODY AXIS  *
C         *                                                  *
C         ****************************************************
C
C     12-8-76      SYSTEM COMMON ENLARGED
      COMMON/SYSTEM/A(1000),L(400)
C
C   VERSION IDENTIFIER
C
      DATA KSEULER /120876/
      EQUIVALENCE(KSEULER ,L(261))
C
C             *** VARIABLE INPUTS ***
C
      EQUIVALENCE(SPHI     ,A( 13)),(CPHI     ,A( 14)),(SECTHT   ,A( 15)),
     1           (TANTHT   ,A( 16)),(P        ,A( 26)),(Q        ,A( 27)),
     2           (R        ,A( 28))
C
C             *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(PHIDT    ,A( 29)),(THETDT   ,A( 30)),(PSIDT    ,A( 31)),
     1           (PD       ,A( 32)),(QD       ,A( 33)),(RD       ,A( 34))
C
C             *** CONSTANTS ***
C
      EQUIVALENCE(RADTOD  ,A(141))
C
C          CALCULATE ATTITUDES
      PHIDT = P+TANTHT*(Q*SPHI+R*CPHI)
      THETDT = Q*CPHI-R*SPHI
      PSIDT = SECTHT*(Q*SPHI+R*CPHI)
C          CALCULATE ANGULAR VELOCITY IN DEGREES
      PD = P*RADTOD
      QD = Q*RADTOD
      RD = R*RADTOD
C
      RETURN
      END
!ASSIGN (M!SI,D3,SILSS)

!FORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

[illegible]

```
      DATA KSILS   /2HSILS/
      EQUIVALENCE(KSILS   ,L(262))
C
C           *** VARIABLE INPUTS ***
C
      EQUIVALENCE(X         ,A(149)),(Y         ,A(150)),(H         ,A(151))
C
      EQUIVALENCE(ILSFLG  ,L(  4)),(JJCOMP  ,L(  8)),(LSILS    ,L(248))
C
C           *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(EGS       ,A(171)),(ELOC      ,A(172)),(HSLOPE   ,A(173)),
     1            (RGS       ,A(174)),(RLOC      ,A(175)),(RRINK    ,A(176))
C
C           *** INTERNALLY GENERATED PARAMETERS ***
C
      EQUIVALENCE(AGS       ,A(177)),(XLOC      ,A(178)),(YLOC     ,A(179)),
     1            (SGS       ,A(180)),(SLOC      ,A(181)),(HRINK    ,A(182)),
     2            (BSLOPE    ,A(183)),(USLOPE    ,A(184)),(XRINK    ,A(185)),
     3            (CILS1     ,A(186)),(CILS2     ,A(187)),(CILS3    ,A(188)),
     4            (CILS4     ,A(189))
C
C           *** CONSTANTS ***
C
      EQUIVALENCE(RADTOD  ,A(141)),(DEGTOR  ,A(142))
C
      IF(JJCOMP.EQ.0)GO TO 99
      IF(LSILS.EQ.1)GO TO 20
C  NO ILS REQUIRED
      EGS=ELOC=0.0
      GO TO 30
   99 CONTINUE
C EXECUTE ONLY IF INITIALISATION NOT COMPLETE
      GO TO (1,2,3)ILSFLG
C     ILSFLG
C       1        MAIN RUNWAY,3 DEG SLOPE
C       2        MAIN RUNWAY,6 DEG SLOPE
C       3        MAIN RUNWAY,2 SEGMENT
```

```
      1 CONTINUE
C     ILS CONSTANTS FOR MAIN RUNWAY,3 DEG GLIDE SLOPE
        XGS      = 1000.0
        XLOC     =11000.0
        YLOC     =     0.0
        SGS      =     0.75
        SLOC     =     2.0
        USLOPE   =     3.0
        BSLOPE   =     3.0
        HKINK    =     0.0
        GO TO 10
      2 CONTINUE
C     ILS CONSTANTS FOR MAIN RUNWAY,6 DEG GLIDE SLOPE
        XGS     =  1000.0
        XLOC    = 11000.0
        YLOC    =     0.0
        SGS     =     0.75
        SLOC    =     2.0
        USLOPE  =     6.0
        BSLOPE  =     6.0
        HKINK   =     0.0
        GO TO 10
      3 CONTINUE
C     ILS CONSTANTS FOR2 SEGMENT (6/3) TO MAIN RUNWAY
        XGS     =  1000.0
        XLOC    = 11000.0
        YLOC    =     0.0
        SGS     =     0.75
        SLOC    =     2.0
        USLOPE  =     6.0
        BSLOPE  =     3.0
        HKINK   =   500.0
     10 CONTINUE
C     COLLECTIVE CONSTANTS
        CILS1 = RADTOD/SGS
        CILS2 = RADTOD/SLOC
        CILS3 = TAN(BSLOPE*DEGTOR)
        CILS4 = TAN(USLOPE*DEGTOR)- CILS3
        XKINK = XGS - HKINK/CILS3
C     MAIN ROUTINE
     20 CONTINUE
C   HERE IMMEDIATELY IF IN 'COMPUTE'
        RKINK = -(X - XKINK)
        RGS   = -(X - XGS)
        RLOC  = -(X - XLOC)
        HSLOPE=RGS*CILS3 + 0.5*(RKINK + ABS(RKINK))*CILS4
        EGS   =BOUND(-1.0,1.0,CILS1*(H - HSLOPE)/RGS)
        ELOC  =BOUND(-1.0,1.0,CILS2*(Y - YLOC)/RLOC)
     30 CONTINUE
        RETURN
        END
IASSIGN (M1S1,03,SINTS)

IFORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE SINIT
C
C          ****************************************************
C          *                                                  *
C          *    WRITTEN BY : D.OLDFIELD                       *
C          *    DATE    17.10.77                              *
C          *    DESCRIPTION : THIS SUBROUTINE CALCULATES      *
C          *      THE INITIAL VALUES OF THE INERTIA COEFFICIENTS *
C          *      GIVEN IX,IY,IZ,IZX                          *
C          *      THE INITIAL VALUES OF VARIABLES LISTED BELOW *
C          *      FROM THE STATED INPUT VARIABLES FROM THE USER *
C          *                                                  *
C          *      USER PROVIDED INFORMATION:-                 *
C          *           BETADIC,ALFADIC,GAMDIC,VKTIC,XIC,YIC,  *
C          *           HIC,ISHR,VWKTO,PSIWD,PSIKDIC,SWREF,SPAN,W *
C          *                                                  *
C          *      INITIAL VALUES OUTPUT:-                     *
C          *           HSLOPE,CPSIW,SPSIW,VWNLO,VWELO,SHRFAC, *
C          *           VWN,VWE,VWD,DENR,SPSNDR,TEMPR,PRESSR,  *
C          *           ROOTSIG,VT,VK,VKNIC,VKEIC,VKDIC,VN,VE, *
C          *           VD,PSIDIC,THETADIC,RADTOD,DEGTOR,      *
C          *           FPSTKT,KTOFPS,G,RHOSL,SPSL,PDIC,QDIC,  *
C          *           RDIC,PHIDIC,XMASS,RXMASS,SB2           *
C          *      N.B                                         *
C          *      HIC NEGATIVE PLACES AIRCRAFT ON ILS BEAM    *
C          *           AT SPECIFIED RANGE XIC                 *
C          *      KISA = 0 (DEFAULT) SELECTS SEA LEVEL ATMOSPHERE *
C          *      VKTIC= 0.0 SETS SYSTEM UP FOR ZERO GROUND SPEED *
C          *           OTHERWISE SETS UP FOR AIRSPEED(EAS)=VKTIC*
C          *                                                  *
C          ****************************************************
C
C      12-8-76      SYSTEM COMMON ENLARGED
C     17.10.77  CI3 CORRECTED
C     8. 2.78 FLAG FOR HIGH SPEED DAC CONTROL ADDED
C     5.6.78 COMMON HSHIO ENLARGED TO 2 WORDS
      COMMON/SYSTEM/A(1000),L(400)
      COMMON/HSHIO/IHSHIO,IDUM
C
C  VERSION IDENTIFIER
C
      DATA KSINIT  /050678/
      EQUIVALENCE(KSINIT  ,L(263))
C
C        ***     INPUT VARIABLES     ***
C
      EQUIVALENCE(XIX       ,A( 48)),(XIY     ,A( 49)),(XIZ     ,A( 50)),
     1           (XIZX      ,A( 51)),(ZCG     ,A( 98)),(XCGREF  ,A( 99)),
     2           (ZCGREF    ,A(100)),(SPAN    ,A(103)),(SWREF   ,A(106)),
     3           (HSLOPE    ,A(173)),(BETADIC ,A(191)),(ALFADIC ,A(192)),
     4           (GAMDIC    ,A(193)),(VKTIC   ,A(194)),(XIC     ,A(195)),
     5           (HIC       ,A(197)),(XCG     ,A(199)),(VWKTO   ,A(202)),
     6           (PSIWD     ,A(203)),(PSIDIC  ,A(204)),(SHRFAC  ,A(228)),
     7           (W         ,A(198))
      EQUIVALENCE                    (ISHR    ,L(  5)),(KISA    ,L(  1))
C
C        ***     OUTPUT VARIABLES    ***
C
      EQUIVALENCE(RXMASS    ,A(  1)),(XMASS   ,A(  2)),(CI1     ,A( 35)),
     1           (CI2       ,A( 36)),(CI3     ,A( 37)),(CI4     ,A( 38)),
     2           (CI5       ,A( 39)),(CI6     ,A( 40)),(CI7     ,A( 41)),
     3           (CI8       ,A( 42)),(CI9     ,A( 43)),(CI10    ,A( 44)),
     4           (VWN       ,A( 55)),(VWE     ,A( 56)),(VWD     ,A( 57)),
     5           (RHOSL     ,A( 79)),(SPSL    ,A( 80)),(DXCG    ,A(101)),
     6           (DZCG      ,A(102)),(SB2     ,A(108)),(X       ,A(149)),
     7           (VKNIC     ,A(211)),(VKEIC   ,A(212)),(VKDIC   ,A(213)),
     8           (PHIDIC    ,A(214)),(THETADIC,A(215)),
     9           (VN        ,A(217)),(VE      ,A(218)),(VD      ,A(219))
C
      EQUIVALENCE(PDIC      ,A(220)),(QDIC    ,A(221)),(RDIC    ,A(222)),
     1           (CPSIW     ,A(224)),(SPSIW   ,A(225)),(VWNLO   ,A(226)),
     2           (VWELO     ,A(227)),(RHO     ,A( 72)),(SPSOND  ,A( 73)),
     3           (ROOTSIG   ,A( 71)),(VT      ,A( 67)),(VK      ,A( 65)),
     4           (TALPHA    ,A( 88)),(PALPHA  ,A( 86))
C
```

```
C         ***    CONSTANTS DERIVED    ***
C
      EQUIVALENCE(RADTOD  ,A(141)),(DEGTOR  ,A(142)),(FPSTKT  ,A(143)),
     1            (XK2FPS  ,A(144)),(G       ,A(145)),(DENR    ,A( 78)),
     2            (SPSNDR  ,A( 84)),(TEMPR   ,A( 81)),(PRESSR  ,A( 82)),
     3            (TEMPSL  ,A( 76)),(PRESSL  ,A( 77)),(RG      ,A(146))
C
C         INITIALISE CONSTANTS
      DATA RADTOD,DEGTOR/57.295780,0.0174533/
      DATA FPSTKT,XK2FPS/0.5921053,1.688889/
      DATA G,RHOSL,SPSL/32.174,0.0023769,1116.45/
      DATA RG/0.03108/
      DATA TEMPSL,PRESSL/288.15,14.69597/
      DATA DENR,SPSNDR,TEMPR,PRESSR/1.0,1.0,1.0,1.0/
      DATA PDIC,QDIC,RDIC,PHIDIC/0.0,0.0,0.0,0.0/
      DATA KISA/0/
      DATA ISHR/1/
C
C  SET FLAG FOR HIGH SPEED HYBRID OUTPUT
      DATA IHSHIO/1/
C
      CIO = 1.0/(XIZ*XIX-XIZX*XIZX)
C         CALCULATE INITIAL VALUES OF INERTIA COEFFICIENTS
      CI1 = CIO*XIZ
      CI2 = CIO*XIZX
      CI3 = CIO*(XIX-XIY+XIZ)*XIZX                                   171077
      CI4 = CIO*(XIZ*(XIY-XIZ)-XIZX*XIZX)
      CI5 = 1.0/XIY
      CI6 = CI5*XIZX
      CI7 = CI5*(XIZ-XIX)
      CI8 = CIO*XIX
      CI9 = CIO*((XIX-XIY)*XIX+XIZX*XIZX)
      CI10 = CIO*XIZX*(-XIX+XIY-XIZ)
C
C        ADJUST H FOR A/C ON BEAM (H<0)
      IF (HIC.GT.0.0) GO TO 10
      X = XIC
      CALL GILS
      HIC = HSLOPE
   10 CONTINUE
C
      IF (KISA.EQ.0) GO TO 24
C        SET UP INITIAL ATMOSPHERIC VALUES
      CALL SATMOS(HIC,DENR,SPSNDR,TEMPR,PRESSR)
   24 ROOTSIG = SQRT(DENR)
      SPSOND = SPSL*SPSNDR
      RHO    = RHOSL*DENR
      TALPHA = TEMPSL*TEMPR
      PALPHA = PRESSL*PRESSR
C
C
C        CALCULATE LOCAL TRUE WIND COMPONENTS GIVEN DATUM WIND
C        SPEED AND DIRECTION
C
      CPSIW = COS(PSIWD*DEGTOR)
      SPSIW = SIN(PSIWD*DEGTOR)
      VWNLO = -VWKTO*XK2FPS*CPSIW
      VWELO = -VWKTO*XK2FPS*SPSIW
C        ALLOWANCE FOR WIND SHEAR
      CALL WSHEAR(ISHR,HIC,SHRFAC)
      VWN = VWNLO*SHRFAC
      VWE = VWELO*SHRFAC
      VWD = 0.0
C
C  SET STARTING VALUE OF TRACK EQUAL TO HEADING
      PSIRDIC = PSIDIC
C  AND STORE DESIRED HEADING FOR REFERENCE
      PSIDICO = PSIDIC
C
C        CALCULATE SIN AND COS OF GAMMA
      CGV = COS(GAMDIC*DEGTOR)
      CGH = COS(PSIRDIC*DEGTOR)
      SGV = SIN(GAMDIC*DEGTOR)
      SGH = SIN(PSIRDIC*DEGTOR)
      IF (VKTIC.LE.1.0) GO TO 20
C
```

```
C             SET UP VELOCITIES FOR DESIRED AIRSPEED
C             AND ITERATE TO GET DESIRED HEADING
C   INITIALISE LOOP COUNT
        I = 0
C   LOOP
     30 CONTINUE
        I = I + 1
        CGH = COS(PSIKDIC*DEGTOR)
        SGH = SIN(PSIKDIC*DEGTOR)
C
        VT = VKTIC*KK2FPS/ROOTSIG
C           SOLVE QUADRATIC FOR VK
        B =-2.0*((VWN*CGH+VWE*SGH)*CGV-VWD*SGV)
        C = VWN*VWN+VWE*VWE+VWD*VWD-VT*VT
        VK = (-B+SQRT(B*B-4.0*C))*0.5
C           CALCULATE GROUND SPEED COMPONENTS
        VKNIC = VK*CGV*CGH
        VKEIC = VK*CGV*SGH
        VKDIC = -VK*SGV
C       CALCULATE COMPONENTS OF AIRSPEED IN EARTH AXES
        VN = VKNIC-VWN
        VE = VKEIC-VWE
        VD = VKDIC-VWD
C       CALCULATE INITIAL HEADING
        PSIAD = 0.0
        IF (VE.EQ.0.0.AND.VN.EQ.0.0) GO TO 16
        PSIAD = ATAN2(VE,VN)*RADTOD
     16 PSIDIC = PSIAD-BETADIC
        DP = PSIDICO - PSIDIC
C   LIMIT NUMBER OF ITERATIONS FOR SAFETY
        IF(I.GE.10) GO TO 26
C   TEST FOR HEADING GOOD ENOUGH
        IF(ABS(DP).LT.0.1) GO TO 26
        PSIKDIC = PSIKDIC + DP
        GO TO 30
C
C             SET UP VALUES FOR ZERO GROUND SPEED
C

     20 VK = 0.0
        VKNIC = 0.0
        VKEIC = 0.0
        VKDIC = 0.0
        VN = -VWN
        VE = -VWE
        VD = -VWD
        VT = SQRT(VN*VN+VE*VE+VD*VD)
C
     26 GAMD = 0.0
        IF (VN.EQ.0.0.AND.VE.EQ.0.0.AND.VD.EQ.0.0) GO TO 18
        GAMD = ATAN2(-VD,SQRT(VN*VN+VE*VE))*RADTOD
     18 THETADIC = GAMD+ALFADIC
C
C             AIRCRAFT RELATED CONSTANTS
C
        SB2 = SWREF*SPAN*SPAN
        XMASS = W/G
        RXMASS = 1.0/XMASS
        DXCG = XCG-XCGREF
        DZCG = ZCG-ZCGREF
        RETURN
        END
!ASSIGN (MISI,D3,SPTHS)

!FORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE SPATH
C
C          *****************************************************
C          *                                                   *
C          *    WRITTEN BY : D.OLDFIELD                        *
C          *    DATE : 12-8-76                                 *
C          *    DESCRIPTION : THIS SUBROUTINE CALCULATES THE   *
C          *      ANGLE OF FLEVATION AND TRACK AND THE VELOCITY *
C          *      OVER THE GROUND                              *
C          *                                                   *
C          *****************************************************
C
C     12-8-76     SYSTEM COMMON ENLARGED
      COMMON/SYSTEM/A(1000),L(400)
C
C   VERSION IDENTIFIER
C
      DATA KSPATH  /120876/
      EQUIVALENCE(KSPATH  ,L(264))
C
C               *** VARIABLE INPUTS ***
C
      EQUIVALENCE(VKN     ,A( 52)),(VKE     ,A( 53)),(VKD     ,A( 54))
C
C               *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(GAMMAR  ,A( 61)),(GAMMAD  ,A( 62)),(PSIKR   ,A( 63)),
     1            (PSIKD  ,A( 64)),(VK      ,A( 65)),(VKKT    ,A( 66))
C
C               *** CONSTANTS ***
C
      EQUIVALENCE(FPSTKT  ,A(143)),(RADTOD  ,A(141))
C
C          CALCULATE VELOCITY OVER GROUND
      VK = SQRT(VKN*VKN+VKE*VKE)
      VKKT = VK*FPSTKT
C
C                    *** CALCULATE FLIGHT PATH ANGLES ***
C
C          INITIALISE ANGLES TO ZERO
      PSIKR = 0.0
      GAMMAR = 0.0
C          CALCULATE ANGLE OF ELEVATION
C              RANGE +90 TO -90
      IF(VKD.EQ.0.0.AND.VK.EQ.0.0) GO TO 4
      GAMMAR = ATAN2(-VKD,VK)
    4 GAMMAD = GAMMAR*RADTOD
C          CALCULATE ANGLE OF TRACK
C              RANGE  0(NORTH) TO 360
      IF(VKE.EQ.0.0.AND.VKN.EQ.0.0) GO TO 5
      PSIKR = ATAN2(VKE,VKN)
C          ADJUST FOR RANGE
      IF(VKE.LT.0.0) PSIKR = 6.283185+PSIKR
    5 PSIKD = PSIKR*RADTOD
C
      RETURN
      END
@ASSIGN (MISI,D3,STVVS)

@FORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE STV
C
C     ****************************************************
C     *                                                  *
C     *  WRITTEN BY : B.N.TOMLINSON                      *
C     *  DATE       : 12-8-76                            *
C     *  REVISED    : 2.9.76 TO USE SYSTEM DISCRETES     *
C     *               6.9.76 WITH NEW TV CONSTANTS       *
C     *                      FOR 700 MODEL               *
C     *                      AND SHIP MOTION             *
C     *               8.9.76 HEIGHT AND BELT PROTECTION  *
C     *               5.10.76 TVCONS INCLUDED IN         *
C     *                      SYSTEM COMMON AND           *
C     *                      CYCLE LOGIC REVISED         *
C     *               24.2.78 PILOT LOCATION CORRECTED   *
C     *  DESCRIPTION: HANDLES TV SLEW AND RESET LOGIC    *
C     *               ALSO CALCULATES TV POSITIONS       *
C     *               N.B.                               *
C     *                NTVB MUST BE PROVIDED BY USER TO  *
C     *                     SELECT APPROPRIATE           *
C     *                     TV CONSTANTS                 *
C     *                                                  *
C     ****************************************************
C
C     LOGIC INPUTS
C       BELT CYCLE     LCYCLE = 1
C       BX RESET       LXTVIC = 1
C       BY RESET       LYTVIC = 1
C       FORWARD SLEW   LFWD   = 1
C       BACKWARD SLEW  LBACK  = 1
C     LOGIC OUTPUTS
C       IDSYNC
C     CONTROL OUTPUTS
C       XIHX
C       XIHY
C     12-8-76       SYSTEM COMMON ENLARGED

      COMMON/SYSTEM/A(1000),L(400)
C
C     VERSION IDENTIFIER
C
      DATA KSTV     /240278/
      EQUIVALENCE(KSTV     ,L(265))
      DIMENSION TVCON(8,3)
      DIMENSION IDS(32),ISLR(16),IP(32)
C
C           *** VARIABLE INPUTS ***
C
      EQUIVALENCE(X        ,A(149)),(Y       ,A(150)),(H       ,A(151)),
     1            (STHETA  ,A( 11)),(CTHETA  ,A( 12)),(SPSI    ,A(  9)),
     2            (CPSI    ,A( 10)),(PSIDT   ,A( 31)),(THETDT  ,A( 30)),
     3            (VKN     ,A( 52)),(VKE     ,A( 53)),(VKD     ,A( 54)),
     4            (DXCG    ,A(101)),(DZCG    ,A(102)),(CREF    ,A(107))
C     EQUIVALENCE (VSHIPKT ,A( 169))                                    240278
      EQUIVALENCE(JJCOMP   ,L(  8))
      EQUIVALENCE(IDS(1)   ,L(223)),(IP(1)   ,L(191))
      EQUIVALENCE(LCYCLE   ,IDS(5)),(LXTVIC  ,IDS(6)),(LYTVIC  ,IDS(7))
      EQUIVALENCE(LFWD     ,ISLR(4)),(LBACK  ,ISLR(5))
C
C           *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(XTV      ,A(152)),(YTV     ,A(153)),(HTV     ,A(154)),
     1            (XDTV    ,A(155)),(YDTV    ,A(156)),(HDTV    ,A(157)),
     2            (XIHX    ,A(159)),(XIHY    ,A(160)),(XDOTM   ,A(158)),
     3            (XIC     ,A(195)),(XPCG    ,A(280)),(ZPCG    ,A(281)),
     4            (RXP     ,A(282)),(RHP     ,A(283)),(XICF    ,A(284))   240278
C
      EQUIVALENCE(IDSYNC,ICLR(16))                                      240278
      EQUIVALENCE(ISLR(1) ,L(151)),(ICLR(1) ,L(171))
C
C           *** CONSTANTS ***
C
      EQUIVALENCE(XP       ,A(109)),(ZP      ,A(110)),(XK2FPS  ,A(144))
```

```
      C
            EQUIVALENCE(NTVB     ,L(  7))
      C
      C         ***  INTERNALLY GENERATED PARAMETERS  ***
      C
            EQUIVALENCE(SXPL     ,A(161)),(SXMIN    ,A(162)),(SXPLUS  ,A(163)),
           1            (SXMINUS ,A(164)),(SYPL     ,A(165)),(SYMIN   ,A(166)),
           2            (HTVLIM  ,A(167)),(XDTMX    ,A(168))
      C
            EQUIVALENCE(TVCON (1,1),A(365))
      C
      C     NTVB = 1(700),2(2000) OR 3(5000),ACCORDING TO BELT USED
      C
      C  SXPL - TV X TRAVEL LIMIT,POSITIVE,AND 'NEAR BELT JOIN'
            DATA TVCON(1,1),TVCON(1,2),TVCON(1,3)/ 11590.0, 33000.0, 77500.0/
      C  SXMIN - TV X TRAVEL LIMIT,NEGATIVE,AND 'NEAR BELT JOIN'
            DATA TVCON(2,1),TVCON(2,2),TVCON(2,3)/-11590.0,-33000.0,-77500.0/
      C  SXPLUS - X CYCLE LIMIT
            DATA TVCON(3,1),TVCON(3,2),TVCON(3,3)/ 12800.0, 34500.0, 89600.0/
      C  SXMINUS - X CYCLE LIMIT
            DATA TVCON(4,1),TVCON(4,2),TVCON(4,3)/-12800.0, 34500.0,-89600.0/
      C  SYPL - TV Y TRAVEL LIMIT,POSITIVE(RIGHT)
            DATA TVCON(5,1),TVCON(5,2),TVCON(5,3)/  2700.0,  7650.0, 54650.0/
      C  SYMIN - TV Y TRAVEL LIMIT,NEGATIVE(LEFT)
            DATA TVCON(6,1),TVCON(6,2),TVCON(6,3)/ -2700.0, -7650.0,-54650.0/
      C  HTVLIM - CEILING FOR TV OPERATION
            DATA TVCON(7,1),TVCON(7,2),TVCON(7,3)/   600.0,  1480.0,  4285.7/
      C  XDTMX - TV SLEW VELOCITY
            DATA TVCON(8,1),TVCON(8,2),TVCON(8,3)/   389.0,   368.0,  2727.3/
      C
      C
            IF(JJCOMP.EQ.1)GO TO 5
      C  EXECUTE ONLY IF INITIALISATION NOT COMPLETE
            SXPL     = TVCON(1,NTVB)
            SXMIN    = TVCON(2,NTVB)
            SXPLUS   = TVCON(3,NTVB)
            SXMINUS  = TVCON(4,NTVB)
            SYPL     = TVCON(5,NTVB)
            SYMIN    = TVCON(6,NTVB)
            HTVLIM   = TVCON(7,NTVB)
            XDTMX    = TVCON(8,NTVB)
      C  STORE XIC FROM ICFILE FOR FUTURE USE
            XICF = XIC
      C  ACTUAL PILOT LOCATION RELATIVE TO CG
            XPCG = XP + DXCG*CREF                                              240278
            ZPCG = ZP - DZCG*CREF                                              240278
      C
          5 CONTINUE
      C  HERE IMMEDIATELY IF IN 'COMPUTE'
      C
      C   **********************************
      C   *                                *
      C   *      SLEW                      *
      C   *                                *
      C   **********************************
      C   CHECK FOR SLEW DEMAND ON BELT DRIVE
      C   IF LFWD = 1,FORWARD SLEW AT XDTMX FT/SEC
            IF(LFWD.EQ.0)GO TO 10
            XDOTM = XDTMX
            GO TO 14
      C   IF LBACK = 1,BACKWARD SLEW AT XDTMX FT/SEC
         10 CONTINUE
            IF(LBACK.EQ.0)GO TO 12
            XDOTM =-XDTMX
            GO TO 14
         12 XDOTM = VKN
      C  ALLOW FOR SHIP VELOCITY
            XDOTM = XDOTM - VSHIPKT*XK2FPS
         14 CONTINUE
      C   **********************************
      C   *                                *
      C   *      TV INTEGRATION CONTROL    *
      C   *                                *
      C   **********************************
      C
      C   TEST TO DETERMINE INTEGRATION STATE OF SX AND SY
```

```
C    PUT SX AND SY IN COMPUTE,DESYNC TO ZERO
        XIMX = XIMY = 1.0
        IDSYNC = 0
C   IF LYTVIC = 1,SY RESETS
        IF(LYTVIC.EQ.0)GO TO 20
        XIMY = 0.0
     20 CONTINUE
C   IF LXTVIC = 1,SX RESETS
        IF(LXTVIC.EQ.0)GO TO 22
        XIMX = 0.0
        XIC = XICF
        GO TO 24
C    CHECK FOR TV NEAR BELT JOIN
     22 IF(SXMIN.LT. X.LT.SXPL)GO TO 24
C    TV NEAR BELT JOIN
C    IF LCYCLE = 1,X COMPUTATION CYCLES BETWEEN SXMINUS AND SXPLUS
        IF(LCYCLE.EQ.0)GO TO 24
        IDSYNC = 1
C    IDSYNC = 1 TO DISCONNECT POSITION ERROR DRIVE TO TV
C   COMMAND IMMEDIATE MOVEMENT IF X SITTING WAY OUT
        IF(X.LT.SXMINUS)GO TO 23
C    WHEN SX REACHES SXPLUS,RESET SX TO SXMINUS
        IF( X.LT.SXPLUS)GO TO 24
     23 CONTINUE
        XIMX = 0.0
         XIC = SXMINUS
     24 CONTINUE
C
C   TV POSITION
C
C   PILOT AT XP,ZP - NH ZP NORMALLY NEGATIVE(!) IF EYE ABOVE DATUM LINE
        RXP    = XPCG*CTHETA + ZPCG*STHETA                                2*0278
        RHP    = XPCG*STHETA - ZPCG*CTHETA                                2*0278
        XTV    = X + RXP*CPSI
        YTV    = Y + RXP*SPSI
        HTV    = H + RHP
C   TV RATES
        XDTV   = XDOTH -RXP*SPSI*PSIDT -RHP*CPSI*THETDT
```

```
        YDTV   = VKE +  RXP*CPSI*PSIDT -RHP*SPSI*THETDT
        HDTV   =-VKD +  RXP*THETDT
C   ON LIMITS,PUT RATES TO ZERO TO KEEP TV DRIVE QUIESCENT
        IF(LCYCLE.EQ.1)GO TO 30
        IF((XTV.LT.SXMIN).OR.(LXTVIC.EQ.1))XDTV = 0.0
     30 CONTINUE
C    IF TV ABOVE CEILING,SET HDTV TO ZERO
        IF(HTV.GT.HTVLIM) HDTV = 0.0
C    KEEP TV POSITION WITHIN LIMITS
        XTV = BOUND(SXMIN,SXPL,XTV)
        YTV = BOUND(SYMIN,SYPL,YTV)
        HTV = BOUND(0.0,HTVLIM,HTV)
C
        RETURN
        END
IASSIGN (MIS1,D3,SVLIS)

IFORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE SVELOC1
C
C        ************************************************************
C        *                                                          *
C        *    WRITTEN BY : D.OLDFIELD                               *
C        *      DATE : 12-8-76                                      *
C        *    REVISED: 22-12-76                                     *
C        *    DESCRIPTION : THIS SUBROUTINE CALCULATES THE          *
C        *      VELOCITY RELATIVE TO BODY AXES GIVEN THE            *
C        *      VELOCITIES RELATIVE TO THE GROUND AND OF            *
C        *      THE WIND                                            *
C        *                                                          *
C        ************************************************************
C
C     12-8-76     SYSTEM COMMON ENLARGED
      COMMON/SYSTEM/A(1000),L(400)
C
C   VERSION IDENTIFIER
C
      DATA KSVELOC1/221276/
      EQUIVALENCE(KSVELOC1,L(266))
C
C             *** VARIABLE INPUTS ***
C
      EQUIVALENCE(VKN     ,A( 52)),(VKE     ,A( 53)),(VKD     ,A( 54)),
     1           (VWN     ,A( 55)),(VWE     ,A( 56)),(VWD     ,A( 57)),
     2           (S11     ,A( 17)),(S12     ,A( 18)),(S13     ,A( 19)),
     3           (S21     ,A( 20)),(S22     ,A( 21)),(S23     ,A( 22)),
     4           (S31     ,A( 23)),(S32     ,A( 24)),(S33     ,A( 25))
C
C             *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(UB      ,A( 58)),(VB      ,A( 59)),(WB      ,A( 60)),
     1           (VN      ,A(217)),(VE      ,A(218)),(VD      ,A(219))
C
C        CALCULATE VELOCITY RELATIVE TO THE AIR
      VN = VKN-VWN
      VE = VKE-VWE
      VD = VKD-VWD
C        CALCULATE VELOCITY RELATIVE TO BODY AXES
      UB = S11*VN+S12*VE+S13*VD
      VB = S21*VN+S22*VE+S23*VD
      WB = S31*VN+S32*VE+S33*VD
      RETURN
      END
!ASSIGN (M:SI,D3,SVL26)

!FORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE SVELOC2
C
C         ***************************************************
C         *                                                 *
C         *    WRITTEN BY : D.OLDFIELD                      *
C         *    DATE : 21.3.78                               *
C         *    DESCRIPTION : THIS SUBROUTINE CALCULATES THE *
C         *      RESULTANT AIR SPEED AND THE EQUIVALENT AIR *
C         *      SPEED,DYNAMIC PRESSURE AND MACH NO.        *
C         *      USING ATMOSPHERIC PROPERTIES APPROPRIATE TO *
C         *      SPECIFIED VALUE OF KISA                    *
C         *                                                 *
C         *        -1 FIXED VALUES APPROPRIATE TO MIC       *
C         *         0 FIXED(SEA LEVEL) VALUES = DEFAULT     *
C         *         1 VALUES VARY WITH HEIGHT               *
C         *                                                 *
C         ***************************************************
C
C     12-8-76      SYSTEM COMMON ENLARGED
C     21.3.78      TRATIO ETC GIVEN DATA VALUES
      COMMON/SYSTEM/A(1000),L(400)
C
C   VERSION IDENTIFIER
C
      DATA KSVELOC2/210378/
      EQUIVALENCE(KSVELOC2,L(267))
C
C             *** VARIABLE INPUTS ***
C
      EQUIVALENCE(UB      ,A( 58)),(VB     ,A( 59)),(WB     ,A( 60)),
     1            (H      ,A(151)),(SWREF  ,A(106)),(CREF   ,A(107)),
     2            (SPAN   ,A(103)),(SB2    ,A(108))
      EQUIVALENCE(KISA    ,L(  1))
C
C             *** VARIABLE OUTPUTS ***
C
      EQUIVALENCE(VT      ,A( 67)),(VTKT   ,A( 68)),(VEAS   ,A( 69)),
     1            (VEASKT ,A( 70)),(QDYN   ,A(125)),(XMACH  ,A( 75)),
     2            (ROOTSIG ,A( 71)),(RHO   ,A( 72)),(SPSOND ,A( 73)),
     3            (DENR   ,A( 78)),(SPSNOR ,A( 84)),(HLFRHOV ,A(124)),
     4            (TEMPR  ,A( 81)),(PRESSR ,A( 82)),(QDYNS  ,A(126)),
     5            (QSCREF ,A(127)),(QSSPAN ,A(128)),(QSB2IV ,A(129)),
     6            (TRATIO ,A( 89)),(ARATIO ,A( 85)),(DRATIO ,A( 83)),
     7            (PRATIO ,A( 87)),(TALPHA ,A( 88)),(PALPHA ,A( 86))
C
C             *** CONSTANTS ***
C
      EQUIVALENCE(FPSTKT  ,B(143)),(RHOSL  ,A( 79)),(SPSL   ,A( 80)),
     1            (TEMPSL ,A( 76)),(PRESSL ,A( 77))
C
C       DATA TRATIO,ARATIO,DRATIO,PRATIO TO AVOID PROBLEMS
C       DURING INITIALISATION
      DATA TRATIO,ARATIO,DRATIO,PRATIO/4*1.0/                          210378
C
C       *****
C
C       CALCULATE TOTAL AIR SPEED
      VT = SQRT(UB*UB+VB*VB+WB*WB)
      VTKT = VT*FPSTKT
C       SET VALUES OF ROOTSIG,RHO AND SPSOND
      IF (KISA.NE.1) GO TO 3
C       CALL SATMOS TO AMMEND RHO,ROOTSIG,SPSOND
      CALL SATMOS(H,DENR,SPSNOR,TEMPR,PRESSR)
      RHO = RHOSL * DENR
      SPSOND = SPSL * SPSNOR
      ROOTSIG = SQRT(DENR)
      TALPHA = TEMPSL*TEMPR
      PALPHA = PRESSL*PRESSR
    3 CONTINUE
C       CALCULATE EQUIVALENT AIRSPEED
      VEAS = VT*ROOTSIG
      VEASKT = VEAS*FPSTKT
C       CALCULATE DYNAMIC PRESSURE
      HLFRHOV = 0.5*RHO*VT
      QDYN = HLFRHOV*VT
```

```
         QDYNS = QDYN*SWREF
         QSCREF = QDYNS*CREF
         QSSPAN = QDYNS*SPAN
         QSB2IV = HLFRHOV*SB2
C              CALCULATE MACH NO.
         XMACH = VT/SPSOND
C  CALCULATE COMPRESSIBLE ADIABATIC FLOW RELATIONSHIPS
C  MACH NUMBER < 1.0
         XX = 1.0 + 0.2*XMACH*XMACH
C  ADIABATIC STATIC/TOTAL TEMPERATURE
         TRATIO = 1.0/XX
C  ADIABATIC SPEED OF SOUND RATIO
         ARATIO = SQRT(TRATIO)
C  ADIABATIC DENSITY RATIO
         DRATIO = ARATIO**5
C  ADIABATIC STATIC/TOTAL PRESSURE
         PRATIO = DRATIO*TRATIO
         RETURN
         END
IASSIGN (MISI,03,SWNDS)

IFORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
         SUBROUTINE SWIND
C
C           ****************************************************
C           *                                                  *
C           * WRITTEN BY : B.N.TOMLINSON                       *
C           * DATE       : 20.9.76                             *
C           * REVISED    : 22.12.76 TO IMPROVE COMM'N WITH GUSTS *
C           *              24.02.78 VWE CORRECTED + FRACG,NGUSTS PROTECTED
C           * DESCRIPTION: THIS SUBROUTINE CALCULATES TOTAL WIND *
C           *    COMPONENTS, INCLUDING TURBULENCE AND WIND SHEAR *
C           *                                                  *
C           ****************************************************
C
         COMMON/SYSTEM/A(1000),L(400)
C  COMMON 'FRN' AND 'INITGUST' COMMUNICATE WITH S/R GUSTS ONLY
         COMMON/FRN/FRACG,RDECAY,NGUSTS
         COMMON/INITGUST/IFLAG,ISR
         EQUIVALENCE(KSWIND  ,L(268))
C
C   VERSION IDENTIFIER
C
         DATA KSWIND/240278/
C
C          *** INPUT VARIABLES ***
C
         EQUIVALENCE(H        ,A(151)),(HIC      ,A(197)),(VKTIC    ,A(194)),
        1            (VWNLO   ,A(226)),(VWELO    ,A(227)),(VWDLO    ,A(223)),
        2            (CPSIW   ,A(224)),(SPSIW    ,A(225)),(FRAMET1  ,A( 92)),
        3            (USIG    ,A(235)),(VSIG     ,A(236)),(WSIG     ,A(237)),
        4            (SFRACG  ,A(278)),(SRDECAY  ,A(279))
C
         DIMENSION IDS(32)
         EQUIVALENCE(ISHR     ,L(  5)),(JJCOMP   ,L(  8)),(NG       ,L(167)),
        1            (LTURB   ,IDS(8)),(IDS(1)   ,L(223)),(LSEED    ,L(168))
C
C          *** OUTPUT VARIABLES ***
C
```

```
      EQUIVALENCE(UTURB   ,A(232)),(VTURB   ,A(233)),(WTURB   ,A(234)),
     1            (VWN    ,A( 55)),(VWE     ,A( 56)),(VWD     ,A( 57)),
     2            (VWNL   ,A(229)),(VWEL    ,A(230)),(VWDL    ,A(231)),
     3            (SHRFAC ,A(228))
C
C      *** CONSTANTS ***
C
      EQUIVALENCE(XK2FPS ,A(144))
C
      DATA NG/4/
      DATA VWDL/0.0/
C SET UP FRACG,RDECAY
      DATA SFRACG,SRDECAY/0.0,0.7/
      DATA LSEED/0/
C CALCULATE WIND AT HEIGHT
      CALL WSHEAR(ISHR,H,SHRFAC)
      VWNL = VWNLO*SHRFAC
      VWEL = VWELO*SHRFAC
C
      IFLAG = JJCOMP
      IF(JJCOMP.EQ.0)GO TO 10
C INITIALISATION COMPLETE
      IF(LTURB.EQ.1) GO TO 20
C TURBULENCE IS NOT REQUIRED
      GO TO 30
   10 CONTINUE
C INITIALISATION
      H = HIC
C PICK UP TURB PARAMETERS FROM SYSTEM VARIABLES
      FRACG = SFRACG
C CHECK ON RANGE
      IF(FRACG.GT.0.99)FRACG = 0.99                                     240278
      RDECAY = SRDECAY
      ISR = LSEED
C SET UP NGUSTS ACCORDING TO INITIAL SPEED
      NGUSTS = NG*(VKTIC*XK2FPS/200.0)
C CHECK ON RANGE
      IF(NGUSTS.LT.1)NGUSTS = 1                                         240278
      NMAX = 1000.0/FRAMET1                                             240278
      IF(NGUSTS.GT.NMAX)NGUSTS = NMAX                                   240278
   20 CONTINUE
C TURB IS REQUIRED
      CALL GUSTS(UG,VG,WG,FRAMET1,H)
      IF(JJCOMP.EQ.0) GO TO 30
C INITIALISATION IS COMPLETE,SO LET TURBULENCE OUT
      UTURB = USIG*UG
      VTURB = VSIG*VG
      WTURB = WSIG*WG
      GO TO 40
   30 CONTINUE
C TURBULENCE OUTPUTS SET TO ZERO(INITIALISATION,OR NOT REQUIRED)
      UTURB = VTURB = WTURB = 0.0
   40 CONTINUE
C     CALCULATE TOTAL WIND COMPONENTS
      VWN = VWNL - (UTURB*CPSIW - VTURB*SPSIW)
      VWE = VWEL - (VTURB*CPSIW + UTURB*SPSIW)                          240278
      VWD = VWDL - WTURB
C
      RETURN
      END
!ASSIGN (MISI,D3,SYSCS)

!FORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE SYSCOM
C
C     ***************************************************
C     *                                                 *
C     *   WRITTEN BY : B.N.TOMLINSON                    *
C     *   DATE : 24.2.78                                *
C     *     THIS ROUTINE CALLS PV200 TO PROVIDE ACCESS  *
C     *      TO SYSTEM COMMON VARIABLES                 *
C     *   REVISED:21.9.76 TO INCLUDE INPUT FROM CHANGES FILE*
C     *          8.10.76 TVCON(8,3) ADDED               *
C     *          29.11.76 DISCRETES NAMES ADDED         *
C     *          22.12.76 FRAMETIME VARIABLES ADDED     *
C     *          31.5.77 NRUN ADDED                     *
C     *          18.10.77 LIDACA ETC.                   *
C     *           8. 2.78 DUMMY 'A' VARIABLES REPLACED BY ARRAYS
C     *           24.2.78 LIDAC REVISED                 *
C     *                                                 *
C     ***************************************************
C
C
C   VERSION IDENTIFIER
C
      DATA KSYSCOM/240278/
C  SET UP COMMUNICATION WITH DAC HARDWARE ADDRESSES
      COMMON/DACNOS/IDACAD(50)                                          080278
C
      COMMON/SYSTEM/
     X HXMASS,XMASS,PSIR,PSID,THETAR,THETAD,PHIR,PHID,SPSI,
     1 CPSI,STHETA,CTHETA,SPHI,CPHI,SECTHT,TANTHT,S11(9),               080278
     2                       P,Q,R,PHIDT,                               080278
     3 THETDT,PSIDT,PD,QD,RD,C11(10),                                   080278
     4                       PDOT,QDOT,RDOT,XIX,XIY,                    080278
     5 XIZ,XIZX,VKN,VKE,VKD,VWN,VWE,VWD,UB,VB,
     6 WB,GAMMAR,GAMMAD,PSIKR,PSIKD,VK,VKKT,VT,VTKT,VEAS,
     7 VEASKT,ROOTSIG,RHO,SPSOND,A74,XMACH,TEMPSL,PRESSL,DENR,RHOSL,
     8 SPSL,TEMPR,PRESSR,DRATIO,SPSNDR,ANATIO,PALPHA,PRATIO,TALPHA,
     C                                                  TRATIO,
     9 A90,DELT1,FRAMET1,A93,TIME,DELT2,FRAMET2,A97,ZCG,XCGREF
C  100
      COMMON/SYSTEM/
     X ZCGREF,DXCG,DZCG,SPAN,XLTAIL,STAIL,SWREF,CREF,SB2,XP,
     1 ZP,ALFAR,ALFAD,BETAR,BETAD,A115,SALFA,CALFA,SBETA,CBETA,
     2 ALFADOT,BETADOT,A122,A123,HLFRHOV,QDYN,QDYNS,QSCREF,QSSPAN,
     C                                                  QSB2IV,
     3 A130,XLLTOT,XMMTOT,XNNTOT,FIX,FTY,FTZ,FIN,FIE,FID,
     4 A140,RADTOD,DEGTOR,FPSTKT,XK2FPS,G,RG   ,A147,A148,X,
     5 Y,H,XTV,YTV,HTV,XDTV,YDTV,HDTV,XDDTM,XIHX,
     6 XIHY,SXPL,SXMIN,SXPLUS,SXMINUS,SYPL,SYMIN,HIVLIM,XDTMX,VSHIPKT,
     7 A170,EGS,ELOC,HSLOPE,RGS,RLOC,RKINK,XGS,XLOC,YLOC,
     8 SGS,SLOC,HKINK,HSLOPE,USLOPE,XKINK,CILS1,CILS2,CILS3,CILS4,
     9 A190,BETADIC,ALFADIC,GAMDIC,VKTIC,XIC,YIC,HIC,W,XCG
C  200
      COMMON/SYSTEM/
     X XILSFLAG,XISHR,VWKTO,PSIWD,PSIDIC ,A205T210(6),                  080278
     1       VKNIC,VKEIC,VKDIC,PHIDIC,THETADIC,A216   ,VN,VE,VD,        080278
     2 PDIC,QDIC,RDIC,VWDLO,CPSIW,SPSIW,VWNLO,VWELO,SHRFAC,VWNL,
     3 VWEL,VWDL,UTURB,VTURB,WTURB,USIG,VSIG,WSIG,VKNDOT,VKEDOT,
     4 VKDDOT,AXCG,AYCG,AZCG,AZCG1,AXACG,AYACG,AZACG,AXP,AYP,
     5 AZP,AZAP,AY1,A22,AX3,AY4,A25,X1,Y1,Z1,
     6 X2,Y2,Z2,X3,Y3,Z3,X4,Y4,Z4,X5,
     7 Y5  ,Z5  ,A272T277(6),                   SFRACG,SRDECAY,         080278
     8 A280T299(20)                                                     080278
C  300
      COMMON/SYSTEM/
     X A300,AADC(64),
     6 TVCON(8,3),
     8 A389T399(11)                                                     080278
C  400
      COMMON/SYSTEM/
     X A400,YADC(64),
     6 A465T499(35)                                                     080278
C   500
      COMMON/SYSTEM/
     X A500,ADAC(64),
     6 A(565:1000)
```

```
C
      COMMON/SYSTEM/
     X KISA,NCPASS,NIPASS,ILSFLG ,ISHR,L6    ,NTVB,JJCOMP,NUMD,
     1 NUMV,NDAC(64),
     7 KSADC,KSDAC,KDSCRT1,KDSCRT2,KISDSCR,
     8 L80,NADC(64)
C  100
      COMMON/SYSTEM/
     4 L145,L146,L147,LSILS,L149,
     5 IAD*SL,ISLR(16),
     6 NG,LSEED,NRUN,                                                   310577
     7 IAD*CL,ICLR(16),
     8 L187,L188,L189,
     9 IDSCFL,IP(32)
C  200
      COMMON/SYSTEM/
     2 IDS(32),
     5 KSACCBOD,KSACCLIN,KSACCROT,KSALFBET,KSCOUNT,
     6 KSDCOS,KSEULER,KSILS,KSINIT,KSPATH,KSTV,KSVELOC1,KSVELOC2,
     C                                          KSWIND,KSYSCOM,
     7 IPCOFL,ICO(32)
C  300
      COMMON/SYSTEM/
     X L303,L304,L1DAC(48),                                             240278
     5 L2DAC(48)                                                        240278
C
C
C   EQUIVALENCES  -  DISCRETES
C
C
      EQUIVALENCE(LFWD     ,ISLR( 4)),
     5            (LBACK    ,ISLR( 5))
C
      EQUIVALENCE(IDSYNC   ,ICLR(16))
C
      EQUIVALENCE(LCYCLE   ,IDS( 5)),
     6            (LXTVIC   ,IDS( 6)),
     7            (LYTVIC   ,IDS( 7)),
     8            (LTJRB    ,IDS( 8))
C
C         FOR CALL TO PV200
      DIMENSION NAME(2)
      DATA NAME/'SYSCOM'/
      NAMELIST
C  DEFINE INPUT UNIT FOR SYSTEM CHANGES FILE
      DATA ISYS/50/
C
C
      CALL PV200(NAME)
      GO TO 3
    2 INPUT(101)
    3 CONTINUE
C  INPUT SEMI-PERMANENT CHANGES FROM SYSTEM INPUT FILE-UNIT ISYS
      REWIND ISYS
      INPUT (ISYS)
      REWIND ISYS
      RETURN
      END
!ASSIGN (MISI,03,SCRIS)

!FORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE DSCRT1
C
C     ************************************************************
C     *                                                          *
C     *  AUTHOR  : B.N.TOMLINSON                                 *
C     *  DATE    : 21.9.76                                       *
C     *  PURPOSE : THIS ROUTINE HANDLES DISCRETES IN FRAME 1.    *
C     *            THE FLAGS IAD*SL,IAD*CL,IDSCFL,IPCOFL         *
C     *            ARE SET EXTERNALLY AND DETERMINE WHICH        *
C     *            ROUTINES ARE TO BE EXECUTED                   *
C     *                                                          *
C     ************************************************************
C
      COMMON/SYSTEM/A(1000),L(400)
C
C   VERSION IDENTIFIER
C
      DATA KDSCRT1 /210976/
      EQUIVALENCE(KDSCRT1 ,L( 77))
C
      EQUIVALENCE(IAD*SL  ,L(150)),(IAD*CL  ,L(170)),
     1           (IDSCFL  ,L(190)),(IPCOFL  ,L(270))
      DATA IAD*SL,IAD*CL,IDSCFL,IPCOFL/1,1,1,0/
C
      IF(IAD*SL.EQ.1)CALL READSLR
      IF(IAD*CL.EQ.1)CALL SETCLR
      IF(IDSCFL.EQ.1)CALL READSCR
      IF(IPCOFL.EQ.1)CALL SETDSCR
      RETURN
      END
IASSIGN (MIS1,D3,SCR2S)

IFORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      SUBROUTINE DSCRT2
C
C     ************************************************************
C     *                                                          *
C     *  AUTHOR  : B.N.TOMLINSON                                 *
C     *  DATE    : 24-8-76                                       *
C     *  PURPOSE : THIS ROUTINE HANDLES DISCRETES IN FRAME 2.    *
C     *            THE FLAGS IAD*SL,IAD*CL,IDSCFL,IPCOFL         *
C     *            ARE SET EXTERNALLY AND DETERMINE WHICH        *
C     *            ROUTINES ARE TO BE EXECUTED                   *
C     *                                                          *
C     ************************************************************
C
      COMMON/SYSTEM/A(1000),L(400)
C
C   VERSION IDENTIFIER
C
      DATA KDSCRT2 /240876/
      EQUIVALENCE(KDSCRT2 ,L( 78))
C
      EQUIVALENCE(IAD*SL  ,L(150)),(IAD*CL  ,L(170)),
     1           (IDSCFL  ,L(190)),(IPCOFL  ,L(270))
C
      IF(IAD*SL.EQ.2)CALL READSLR
      IF(IAD*CL.EQ.2)CALL SETCLR
      IF(IDSCFL.EQ.2)CALL READSCR
      IF(IPCOFL.EQ.2)CALL SETDSCR
      RETURN
      END
IASSIGN (MIS1,D3,ISDCS)

IFORTRAN LS,NS
EXT. FORTRAN IV, VERSION E00
```

```
      FUNCTION ISDSCR(ISET,LINE)
C
C     ***********************************************************
C     *                                                         *
C     *  AUTHOR   : B.N.TOMLINSON                               *
C     *  DATE     :  19-8-76                                    *
C     *  PURPOSE  : THIS FUNCTION READS SIGMA DISCRETES IN I.E. *
C     *             PATCHABLE AND DEDICATED DESK SWITCHES.      *
C     *             IT IS INTENDED FOR EXECUTION DURING 'INITIAL'. *
C     *  ARGUMENTS :                                            *
C     *             ISET  SELECTS PATCHABLE DISCRETES(0) OR     *
C     *                         DEDICATED DESK SWITCHES(1).     *
C     *             LINE  DEFINES ELEMENT IN ARRAY              *
C     *             ISDSCR IS RETURNED 0 OR 1                   *
C     *                                                         *
C     ***********************************************************
      COMMON/SYSTEM/A(1000),L(400)
C
C  VERSION IDENTIFIER
C
      DATA KISDSCR /190876/
      EQUIVALENCE(KISDSCR ,L( 79))
      DIMENSION IP(32),IDS(32)
      EQUIVALENCE(ISCR(1) ,L(151)),(ICLR(1) ,L(171)),
     1           (IP(1)   ,L(191)),(IDS(1)  ,L(223)),(ICD(1)  ,L(271))
C
      CALL READSCR
C
      ISDSCR= IP(LINE)
      IF(ISET.EQ.1)ISDSCR= IDS(LINE)
      RETURN
      END
IFIN
```

Appendix C

LISTING OF SL1 PROGRAM - SINGLE LOOP

```
       PROGRAM GENERAL MODEL
      C  SET VERSION NUMBER
            DATA IVERSION/080278/
      C  12.8.76 SYSTEM COMMON ENLARGED
      C  25.8.76 DISCRETES INCLUDED
      C  5.10.76 FRAME TIME VARIABLES TIDIED UP
      C  21.12.76 TOTM AND TOTF SEPARATED AND SOURCE CLEANED UP
      C  8. 2.78 SHORT NAME LIST SPECIFIED TO CUT STORAGE
      C
          CONTROL
            MODE = REALT
          ENDC
       INITIAL
      C
      C  DIMENSION,COMMON,DATA DECLARATIONS
      C
            COMMON/SYSTEM/A(1000),L(400)
            DIMENSION RICVALS(20)
            DATA NIPASS/1/
            EQUIVALENCE(RICVALS(1),A(191))
      C
      C     COMMUNICATION BETWEEN 'INITIAL' AND SYSTEM COMMON
      C
            EQUIVALENCE(VKNIC    ,A(211)),(VKEIC    ,A(212)),(VKDIC   ,A(213)),..
                       (XIC      ,A(195)),(YIC      ,A(196)),(HIC     ,A(197)),..
                       (PDIC     ,A(220)),(QUIC     ,A(221)),(RDIC    ,A(222)),..
                       (PHIDIC   ,A(214)),(THETADIC,A(215)),(PSIDIC  ,A(204)),..
                       (FRAMET1  ,A( 92)),..
                       (DEGTOR   ,A(142)),(DELT1    ,A( 91))
            EQUIVALENCE(NCPASS   ,L(  2)),(NIPASS   ,L(  3)),(ILSFLG ,L(  4)),..
                       (ISHR     ,L(  5))                                         080278
      C
      C  REAL TIME PARAMETERS
            DATA FRAMET1/60.0/
            DATA TIMLEFT1/0.0/
            DATA DELT1/0.050/
      C  NUMBER OF IC VALUES
            DATA NVALS/14/
```

```
      C
      C  SET UP COMMUNICATION,TO ENABLE VARIABLES IN SL1 LABELLED
      C  COMMON TO BE ACCESSED
      C
            DIMENSION NAME(2)
            DATA NAME/'MODEL'/
            NAMELIST CINT1,IALG1,JALG1,IHX,IHY,ITP,IVERSION,MODE,..           080278
               NS1,NVALS,SX,SY,HH,SXIC,SYIC,SHIC,T,TIMLEFT1                   080278
       LABC:CALL PV200(NAME)
            GO TO LABO
       LABB:INPUT(101)
       LABO:CONTINUE
            CALL SYSCOM
            CALL USERCOM
      C
      C  CHECK ACCESS 'ON' AND INITIALISE AD4
            CALL INTLAD4
      C  READ IC CONDITIONS FROM MASTER FILE(F:1),PLACE IN SYSTEM COMMON
            CALL RDICFILE(NVALS,RICVALS)
            ILSFLG = RICVALS(10)
            ISHR   = RICVALS(11)
      C   ***************************
      C   *  INPUT RETAINED CHANGES  *
      C   ***************************
            CALL PV300
      C  INTERPRETER
      C  ***********
            ITP = 1
      C  CALL PV100 IF DESK SWITCH 1 ON
            IF(ISDSCH(1,1).EQ.1)CALL PV100
      C
      C  STEP SIZE (FOR REAL TIME, = FRAMET1/1000.0)
      C  SL1 CALCULATES INTEGRATION STEP FROM CINT/NS.
      C  FOR USER ACCESS,SET CINT FROM DELT
            CINT1 = DELT1
      C   INITIALISE SYSTEM FUNCTIONS.
            CALL SINIT
      C  INTERPRETER
```

```
      C  ***********
            ITP = 2
      C  CALL PV100 IF DESK SWITCH 1 ON
            IF(ISDSCR(1,1).EQ.1)CALL PV100
      C
      C  SET UP INITIAL VALUES FOR INTEGRATIONS
      C
            VVKNIC = VKNIC
            VVKEIC = VKEIC
            VVKDIC = VKDIC
            SXIC   = XIC
            SYIC   = YIC
            SHIC   = HIC
            PPIC   = PDIC*DEGTOR
            QQIC   = QDIC*DEGTOR
            RRIC   = RDIC*DEGTOR
            PPHIC  = PHIDIC*DEGTOR
            TTHIC  = THETADIC*DEGTOR
            PPSIC  = PSIDIC*DEGTOR
      C
      C   FOR CONTROL OF 'POST-INTEGRATION' CALCULATIONS,SET UP VALUE OF NIPASS TO
      C   SUIT INTEGRATION ALGORITHM IALGOR.DEFAULT IS NIPASS=1
            IF(IALG1.EQ.4)NIPASS = 2
            IF(IALG1.EQ.5)NIPASS = 4
      C   TO ENSURE THAT SUCH 'POST-INTEGRATION' ROUTINES ARE CALLED DURING THE
      C   INITIALISATION PASS THROUGH 'Z0001'
            NCPASS=4
      C
      C  INTERPRETER
      C  ***********
            ITP = 3
      C  CALL PV100 IF DESK SWITCH 1 ON
            IF(ISDSCR(1,1).EQ.1)CALL PV100
      C
            INIT = 0
       INIT2:CONTINUE
            INIT = INIT + 1
       END

      C  INITIALISE TWICE SO THAT 'QDYN' ETC SET UP CORRECTLY
            IF(INIT.EQ.1) GO TO INIT2
      C
      C  INTERPRETER
      C  ***********
            ITP = 4
      C  CALL PV100 IF DESK SWITCH 1 ON
            IF(ISDSCR(1,1).EQ.1)CALL PV100
      C
      C  START CLOCK FOR REAL TIME
            IF(MODE) CALL STARTC
       DYNAMIC
      C
      C     *****************************    DYNAMIC     *******************
      C
            IF(MODE) CALL EXITS
       DERIVATIVE LOOP1
      C
      C     *****************************    DERIVATIVE  *******************
      C
      C
      C   ************************
      C   *                      *
      C   *   SYSTEM FUNCTIONS   *
      C   *                      *
      C   ************************
      C  COMMON STATEMENT TO ALLOW CONTROL PARAMETER 'MODE'
      C  TO BE ACCESSED BY THE DERIVATIVE SECTION
            COMMON/Z9980/MODE
            DECLARE LOGICAL MODE
      C  INTEGRATION CONTROL STATEMENTS
            VARIABLE     T = 0.0
            CINTERVAL CINT1 = 0.05
            NSTEPS    NS1   = 1
            ALGORITHM IALG1 = 5 , JALG1 = 5
            RT(MIL,FRAMET1,IFLAG1,TIMLEFT1,1.0,1)
           PROCEDURAL
      C     12-8-76     SYSTEM COMMON ENLARGED
```

```
      COMMON/SYSTEM/A(1000),L(400)
C
C         ** INPUT VARIABLES **
C
      EQUIVALENCE(PHIDT   ,A( 29)),(THETDT  ,A( 30)),(PSIDT,A( 31)),...
                 (PDOT    ,A( 45)),(QDOT    ,A( 46)),(RDOT ,A( 47)),...
                 (VKNDOT  ,A(238)),(VKEDOT  ,A(239)),(VKDDOT,A(240)),...
                 (XIC     ,A(195)),(XDOTM   ,A(158)),(DELT1,A( 91)),...
                 (XIMX    ,A(159)),(XIMY    ,A(160))
C
      EQUIVALENCE(NCPASS  ,L(  2)),(NIPASS  ,L(  3))
C
C         ** OUTPUT VARIABLES **
C
      EQUIVALENCE(PHIR    ,A(  7)),(THETAR  ,A(  5)),(PSIR ,A(  3)),...
                 (P       ,A( 26)),(Q       ,A( 27)),(R    ,A( 28)),...
                 (VKN     ,A( 52)),(VKE     ,A( 53)),(VKD  ,A( 54)),...
                 (X       ,A(149)),(Y       ,A(150)),(H    ,A(151)),...
                 (TIME    ,A( 96))
    END
C                   *********************
C                   *                   *
C                   *PRE-INTEGRAL CALLS *
C                   *                   *
C                   *********************
C
C  READ AND WRITE DISCRETE LINES
      CALL DSCRT1
C
      CALL SCOUNT
C
 PROCEDURAL
      DATA IFIRST/1/
      IF(IFIRST.NE.1) GO TO NOTF
C    INSERT CALLS HERE FOR ROUTINES TO BE EXECUTED AFTER ADC BUT BEFORE INTEG
C    -RATION ,AND WHICH ARE NOT PART OF EVALUATION LOOP FOR DERIVATIVES.
C  ONLY EXECUTE THESE ROUTINES ONCE PER FRAME
C     1) PRE DERIVATIVE S/R CALLS
C
      CALL CONTROLS
C  CALCULATE WIND
      CALL SWIND
C
C  RESET FLAG
      IFIRST = 0
 NOTF:CONTINUE
      END
C
    PROCEDURAL( = ADC(1))
    END
C
C                   ********************
C                   *                  *
C                   *  ANGULAR MOTION  *
C                   *                  *
C                   ********************
C
C  ATTITUDE ANGLES
      PPSI = INTEG(PPSIDT,PPSIC)
      TTHET = INTEG(TTHDT,TTHIC)
      PPHI = INTEG(PPHIDT,PPHIC)
C
    PROCEDURAL(=PPSI,TTHET,PPHI)
C   EQUATE VARIABLES
      PSIR = PPSI
      THETAR = TTHET
      PHIR = PPHI
C   CALCULATE DIRECTION COSINES
C
      CALL SDCOS
C
    END
C
      PP = INTEG(PPDOT,PPIC)
```

```
      QQ = INTEG(QQDOT,QQIC)
      RR = INTEG(RRDOT,RRIC)
C
      PROCEDURAL(PPSIDT,TTHDT,PPHIDT=PP,QQ,RR)
C     EQUATE VARIABLES
        P = PP
        Q = QQ
        R = RR
C   CALCULATE ATTITUDE RATES
        CALL SEULER
        PPSIDT = PSIDT
        TTHDT = THETDT
        PPHIDT = PHIDT
      END
C
C                  ********************
C                  *                  *
C                  *  LINEAR MOTION   *
C                  *                  *
C                  ********************
C
C     VELOCITIES
        VVKN = INTEG(VVKNDT,VVKNIC)
        VVKE = INTEG(VVKEDT,VVKEIC)
        VVKD = INTEG(VVKDDT,VVKDIC)
C
      PROCEDURAL(=VVKN,VVKE,VVKD)
C     EQUATE VARIABLES
        VKN = VVKN
        VKE = VVKE
        VKD = VVKD
C     CALCULATE BODY AXES VELOCITIES
        CALL SVELOC1
C     CALCULATE ALPHA AND BETA AND ASSOCIATED SINES/COSINES
        CALL SALFBET(DELT1)
      END
      PROCEDURAL(HHDOT,SYDOT=)
C
C     EQUATE VARIABLES
        HHDOT = -VKD
        SYDOT = VKE
        SXDOTM = XDOTM
      END
C
C     CALCULATE POSITIONS
        SX = MODINT(SXDOTM,SXIC,1.0,IHX,T)
        SY = MODINT(SYDOT,SYIC,1.0,IHY,T)
        HH = INTEG(HHDOT,SHIC)
C
      PROCEDURAL
C     USERS ROUTINES
C     CALCULATE TOTAL FORCES IN BODY AXES
C
        CALL TOTF
C
      END
C
      PROCEDURAL(PPDOT,QQDOT,RRDOT = PP,QQ,RR)
C   USER'S ROUTINE TO CALCULATE MOMENTS
        CALL TOTM                                                  080278
C
C   CALCULATE ANGULAR ACCELERATIONS
        CALL SACCROT                                               080278
C   EQUATE VARIABLES
        PPDOT = PDOT                                               080278
        QQDOT = QDOT                                               080278
        RRDOT = RDOT                                               080278
      END
C
C
      PROCEDURAL(VVKNDT,VVKEDT,VVKDDT=)
C     CALCULATE LINEAR ACCELERATIONS
        CALL SACCLIN
C     EQUATE VARIABLES
        VVKNDT = VKNDOT
        VVKEDT = VKEDOT
```

```
      VVKDOT = VKDDOT
     END
    PROCEDURAL( = SX,SY,HH)
C                   ********************
C                   *                  *
C                   *POST-INTEGRAL CALLS*
C                   *                  *
C                   ********************
       NCPASS=NCPASS+1
C   NCPASS = COUNT OF NUMBER OF PASSES EXECUTED SO FAR IN ONE INTEGRATION STEP
C   NIPASS = CONSTANT,SET UP IN INITIAL,FOR NUMBER OF PASSES USED BY
C   INTEGRATION ROUTINE.
       IF(NCPASS.LT.NIPASS)GO TO SKPASS
C   INSERT HERE CALLS TO 'POST-INTEGRATION' ROUTINES,TO BE EXECUTED ONLY
C   ONCE PER STEP.
C
C  FIRST PICK UP TIME
       TIME = T
C   EQUATE VARIABLES
       X = SX
       Y = SY
       H = HH
C   CALCULATE BODY AXES ACCELERATIONS FOR MOTION ('OUTPUT' ONLY)
       CALL SACCBOD
C
C   CALCULATE INSTRUMENT READINGS ('OUTPUT' ONLY)
C   CALCULATE ILS ('OUTPUT' ONLY)
       CALL SILS
C
C   LOGIC AND POSITION FOR VFA CONTROL ('OUTPUT' ONLY)
       CALL STV
C  EQUATE VARIABLES
       SXIC = XIC
       IHX = XIHX
       IHY = XIHY
       SXDOTH = XDOTH
C   CALCULATE EAS,MACH,DYNAMIC PRESSURE ETC.
       CALL SVELOC2

C   CALCULATE FLIGHT PATH ANGLES
       CALL SPATH
C
C      3) POST DERIVATIVE S/R CALLS
C
C      S/R 'OUTSR' HANDLES OUTPUT FUNCTIONS
C
       CALL OUTSR
C
       CALL SDAC
C
C  SET 'FIRST TIME' FLAG ON,READY FOR NEXT TIME ROUND
       IFIRST = 1
    SKPASS : CONTINUE
    END
C
    PROCEDURAL(DAC( 1),DAC( 2),DAC( 3),DAC( 4),DAC( 5),...
               DAC( 6),DAC( 7),DAC( 8),DAC( 9),DAC(10),...
               DAC(11),DAC(12),DAC(13),DAC(14),DAC(15),...
               DAC(16),DAC(17),DAC(18),DAC(19),DAC(20),...
               DAC(21),DAC(22),DAC(23),DAC(24),DAC(25),...
               DAC(26),DAC(27),DAC(28),DAC(29),DAC(30),...
               DAC(31),DAC(32),DAC(33),DAC(34),DAC(35),...
               DAC(36),DAC(37),DAC(38),DAC(39),DAC(40),...
               DAC(41),DAC(42),DAC(43),DAC(44),DAC(45),...
               DAC(46),DAC(47),DAC(48)=)
    END
C
C  PROCESS AUTOMATIC HOLD
       CALL PV700
       IF(MODE) CALL TERMNATE('AIRCRAFT')
   END
   END
   TERMINAL
   END
   END
```

Appendix D

INDEX TO SYSTEM COMMON, REAL VARIABLES

D.1 Numeric order

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 1

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
1	RXMASS		1/XMASS		SINIT	SACCLIN
2	XMASS		AIRCRAFT MASS	SLUG	SINIT	SINIT
3	PSIR		HEADING ANGLE	RADIANS	DER'TIVE	SDCOS
4	PSID		HEADING ANGLE	DEGREES	SDCOS	
5	THETAR		PITCH ATTITUDE	RADIANS	DER'TIVE	SDCOS
6	THETAD		PITCH ATTITUDE	DEGREES	SDCOS	
7	PHIR		BANK ANGLE	RADIANS	DER'TIVE	SDCOS
8	PHID		BANK ANGLE	DEGREES	SDCOS	
9	SPSI		SIN(PSIR)		SDCOS	STV SDCOS
10	CPSI		COS(PSIR)		SDCOS	STV SDCOS
11	STHETA		SIN(THETAR)		SDCOS	STV SDCOS
12	CTHETA		COS(THETAR)		SDCOS	STV SDCOS

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 2

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
13	SPHI		SIN(PHIR)		SDCOS	SEULER SDCOS
14	CPHI		COS(PHIR)		SDCOS	SEULER SDCOS
15	SECTHT		SEC(THETAR)		SDCOS	SEULER SDCOS
16	TANTHT		TAN(THETAR)		SDCOS	SEULER
17	B11		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
18	B12		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
19	B13		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN SACCBOD
20	B21		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
21	B22		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
22	B23		DIRECTION COSINE		SDCOS	SVELOC1

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 3

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SACCLIN SACCBOD
23	B31		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
24	B32		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
25	B33		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN SACCBOD
26	P		ANG VEL,ROLL,BODY AXES	RADS/SEC	DER'TIVE	SEULER SACCROT SACCBOD
27	Q		ANG VEL,PITCH,BODY AXES	RADS/SEC	DER'TIVE	SEULER SACCROT SACCBOD
28	R		ANG VEL,YAW,BODY AXES	RADS/SEC	DER'TIVE	SEULER SACCROT SACCBOD
29	PHIDT		ATTITUDE RATE,BANK	RAD/SEC	SEULER	DER'TIVE
30	THETDT		ATTITUDE RATE,PITCH	RAD/SEC	SEULER	STV

SYSTEM COMMON - REAL VARIABLES 15.3.78

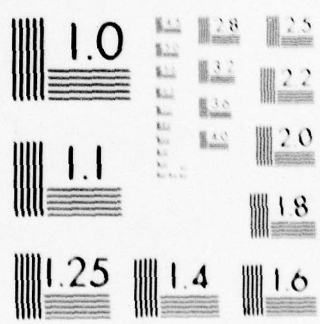
NUMERIC ORDER 10:45 MAR 13,'79 PAGE 4

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						DER'TIVE
31	PSIDT		ATTITUDE RATE,HEADING	RAD/SEC	SEULER	STV DER'TIVE
32	PD		ANG VEL,ROLL,BODY AXES	DEGS/SEC	SEULER	
33	QD		ANG VEL,PITCH,BODY AXES	DEGS/SEC	SEULER	
34	RD		ANG VEL,YAW,BODY AXES	DEGS/SEC	SEULER	
35	CI1		INERTIA COEFFICIENT		SINIT	SACCROT
36	CI2		INERTIA COEFFICIENT		SINIT	SACCROT
37	CI3		INERTIA COEFFICIENT		SINIT	SACCROT
38	CI4		INERTIA COEFFICIENT		SINIT	SACCROT
39	CI5		INERTIA COEFFICIENT		SINIT	SACCROT SINIT
40	CI6		INERTIA COEFFICIENT		SINIT	SACCROT
41	CI7		INERTIA COEFFICIENT		SINIT	SACCROT
42	CI8		INERTIA COEFFICIENT		SINIT	SACCROT
43	CI9		INERTIA COEFFICIENT		SINIT	SACCROT

AD-A074 758 ROYAL AIRCRAFT ESTABLISHMENT FARNBOROUGH (ENGLAND) F/G 9/2
SESAME-A SYSTEM OF EQUATIONS FOR THE SIMULATION OF AIRCRAFT IN --ETC(U)
JAN 79 B N TOMLINSON
UNCLASSIFIED RAE-TR-79008 DRIC-BR-68659 NL

2 OF 2
AD
A074758

END
DATE
FILMED
11-79
DDC


1.0
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 5

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
44	C110		INERTIA COEFFICIENT		SINIT	SACCROT
45	PDOT		ANGULAR ACCN. IN BODY AXES	RADS/SEC2	SACCROT	SACCBOD
46	QDOT		ANGULAR ACCN.IN BODY AXES	RADS/SEC2	SACCROT	SACCBOD
47	RDOT		ANGULAR ACCN. IN BODY AXES	RADS/SEC2	SACCROT	SACCBOD
48	XIX		MOMENT OF INERTIA,ROLL	SLUG FT2	USER	SINIT
49	XIY		MOMENT OF INERTIA,PITCH	SLUG FT2	USER	SINIT
50	XIZ		MOMENT OF INERTIA,YAW	SLUG FT2	USER	SINIT
51	XIZX		MOMENT OF INERTIA,PRODUCT	SLUG FT2	USER	SINIT
52	VKN		VELOCITY REL TO GROUND,NORTH	FT/SEC	DER'TIVE	SVELOC1 SPATH STV
53	VKE		VELOCITY REL TO GROUND,EAST	FT/SEC	DER'TIVE	SVELOC1 SPATH STV
54	VKD		VELOCITY REL TO GROUND,DOWN	FT/SEC	DER'TIVE	SVELOC1 SPATH STV

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 6

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
55	VWN		VELOCITY OF WIND,NORTH	FT/SEC	SWIND	SVELOC1
56	VWE		VELOCITY OF WIND,EAST	FT/SEC	SWIND	SVELOC1
57	VWD		VELOCITY OF WIND,DOWN	FT/SEC	SWIND	SVELOC1
58	UB		VEL COMP REL TO AIR,BODY AXES	FT/SEC	SVELOC1	SVELOC2 SALFBET
59	VB		VEL COMP REL TO AIR,BODY AXES	FT/SEC	SVELOC1	SVELOC2 SALFBET
60	WB		VEL COMP REL TO AIR,BODY AXES	FT/SEC	SVELOC1	SVELOC2 SALFBET
61	GAMMAR		FLIGHT PATH ANGLE,CLIMB	RADIANS	SPATH	SPATH
62	GAMMAD		FLIGHT PATH ANGLE,CLIMB	DEGREES	SPATH	
63	PSIKR		FLIGHT PATH ANGLE,TRACK	RADIANS	SPATH	SPATH
64	PSIKD		FLIGHT PATH ANGLE,TRACK	DEGREES	SPATH	
65	VK		VELOCITY RELATIVE TO GROUND	FT/SEC	SPATH	SINIT SPATH
66	VKKT		VELOCITY RELATIVE TO GROUND	KNOTS	SPATH	

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 7

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
67	VT		TOTAL AIR SPEED	FT/SEC	SVELOC2	SINIT SVELOC2
68	VTKT		TOTAL AIR SPEED	KNOTS	SVELOC2	
69	VEAS		EQUIVALENT AIR SPEED	FT/SEC	SVELOC2	SVELOC2
70	VEASKT		EQUIVALENT AIR SPEED	KNOTS	SVELOC2	
71	ROOTSIG		SQRT(DENSITY RATIO)		SVELOC2	SINIT SVELOC2
72	RHO		AIR DENSITY	SLUG/FT3	SVELOC2	SVELOC2
73	SPSOND		SPEED OF SOUND	FT/SEC	SVELOC2	SVELOC2
74						
75	XMACH		MACH NUMBER		SVELOC2	SVELOC2
76	TEMPSL		AMB TEMP AT SL(STANDARD DAY)	DEG. K	SINIT	SVELOC2 SINIT
77	PRESSL		AMBIENT PRESSURE AT SEA LEVEL	LB/IN 2	SINIT	SVELOC2 SINIT
78	DENR		DENSITY RATIO		SINIT	SVELOC2

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 8

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SINIT
79	RHOSL		DENSITY AT SEA LEVEL	SLUG /FT3	SINIT	SVELOC2 SINIT
80	SPSL		SPEED OF SOUND AT SEA LEVEL	FT/SEC	SINIT	SVELOC2 SINIT
81	TEMPR		ATMOSPHERIC TEMP RATIO		SATMOS	SVELOC2 SINIT
82	PRESSR		ATMOSPHERIC PRESSURE RATIO		SATMOS	SVELOC2 SINIT
83	DRATIO		ADIABATIC DENSITY RATIO		SVELOC2	SVELOC2
84	SPSNDR		SPEED OF SOUND RATIO		SATMOS	SVELOC2 SINIT
85	ARATIO		ADIABATIC SPEED OF SOUND RATIO		SVELOC2	SVELOC2
86	PALPHA		FREE STREAM STATIC PRESSURE	LB/IN 2	SVELOC2	
87	PRATIO		ADIABATIC STATIC/TOTAL PRESS		SVELOC2	
88	TALPHA		FREE STREAM STATIC TEMPERATURE	DEG. K	SVELOC2	
89	TRATIO		ADIABATIC STATIC/TOTAL TEMP		SVELOC2	SVELOC2

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 9

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
90						
91	DELT1		INTEG. STEP LENGTH NO. 1	SECS	USER	INITIAL
92	FRAMET1		FRAME TIME,LOOP 1	MSEC	USER	INITIAL SWIND
93						
94	TIME		TIME	SEC	DER'TIVE	
95	DELT2		INTEG. STEP LENGTH NO. 2	SECS	USER	INITIAL
96	FRAMET2		FRAME TIME,LOOP 2	MSEC	USER	INITIAL
97						
98	ZCG		Z C.G. LOCATION		USER	SINIT
99	XCGREF		X LOCATION OF REF PT		USER	SINIT
100	ZCGREF		Z LOCATION OF REF PT		USER	SINIT
101	DXCG		DIST OF C.G. AHEAD OF M.R.C.		SINIT	STV
102	DZCG		DIST OF C.G. BELOW M.R.C.		SINIT	STV
103	SPAN		WING SPAN	FT	USER	SVELOC2

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 10

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SINIT
104	XLTAIL		TAIL ARM FROM M.R.C.	FT	USER	
105	STAIL		TAIL PLANE AREA	FT2	USER	
106	SWREF		WING REFERENCE AREA	FT2	USER	SVELOC2 SINIT
107	CREF		REFERENCE CHORD	FT	USER	SVELOC2 STV
108	SB2		SWREF*SPAN*SPAN	FT4	SINIT	SVELOC2
109	XP		X LOC OF PILOT REL TO REFCG	FT	USER	STV
110	ZP		Z LOC OF PILOT REL TO REFCG	FT	USER	STV
111	ALFAR		ANGLE OF ATTACK	RADIANS	SALFBET	SALFBET
112	ALFAD		ANGLE OF ATTACK	DEGREES	SALFBET	
113	BETAR		ANGLE OF SIDESLIP	RADIANS	SALFBET	SALFBET
114	BETAD		ANGLE OF SIDESLIP	DEGREES	SALFBET	
115						
116	SALFA		SINE OF ALFA		SALFBET	

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 11

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
117	CALFA		COS OF ALFA		SALFBET	
118	SBETA		SINE OF BETA		SALFBET	
119	CBETA		COS OF BETA		SALFBET	
120	ALFADOT		RATE OF ANGLE OF ATTACK	RADS/SEC	SALFBET	
121	BETADOT		RATE OF ANGLE OF SIDESLIP	RADS/SEC	SALFBET	
122						
123						
124	HLFRHOV		0.5*RHO*VT		SVELOC2	SVELOC2
125	QDYN		DYNAMIC PRESSURE	LB/FT2	SVELOC2	SVELOC2
126	QDYNS		QDYN*SWREF	LB	SVELOC2	SVELOC2
127	QSCREF		QDYNS*CREF	LB-FT	SVELOC2	
128	QSSPAN		QDYNS*SPAN	LB-FT	SVELOC2	
129	QSB2IV		HLFRHOV*SB2		SVELOC2	
130						

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 12

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
131	XLLTOT		TOTAL MOMENT,ROLL,BODY AXES	LB-FT	USER	SACCROT
132	XMMTOT		TOTAL MOMENT,PITCH,BODY AXES	LB-FT	USER	SACCROT
133	XNNTOT		TOTAL MOMENT,YAW,BODY AXES	LB-FT	USER	SACCROT
134	FTX		TOTAL FORCE COMP. IN BODY AXES	LB	USER	SACCLIN SACCBOD
135	FTY		TOTAL FORCE COMP. IN BODY AXES	LB	USER	SACCLIN SACCBOD
136	FTZ		TOTAL FORCE COMP. IN BODY AXES	LB	USER	SACCLIN SACCBOD
137	FTN		TOTAL FORCE COMP IN EARTH AXES	LB	SACCLIN	SACCLIN
138	FTE		TOTAL FORCE COMP IN EARTH AXES	LB	SACCLIN	SACCLIN
139	FTD		TOTAL FORCE COMP IN EARTH AXES	LB	SACCLIN	SACCLIN
140						
141	RADTOD		CONVERSION RADIANS TO DEGREES	CONSTANT	SINIT	SDCOS SPATH SALFBET SEULER

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 13

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SILS SINIT
142	DEGTOR		CONVERSION DEGREES TO RADIANS	CONSTANT	SINIT	SILS SINIT
143	FPSTKT		CONVERSION FT/SEC TO KNOTS	CONSTANT	SINIT	SVELOC2 SPATH
144	KK2FPS		CONVERSION KNOTS TO FT/SEC	CONSTANT	SINIT	SWIND STV SINIT
145	G		ACCELERATION DUE TO GRAVITY	CONSTANT	SINIT	SACCLIN SINIT
146	RG		RECIPROCAL OF G		SINIT	SACCBOD
147						
148						
149	X		X POSITION OF C.G.	FT	DER'TIVE	STV SILS
150	Y		Y POSITION OF C.G.	FT	DER'TIVE	STV SILS
151	H		HEIGHT OF CG	FT	DER'TIVE	SATMOS

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 14

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SVELOC2 STV SILS SWIND
152	XTV		X POSITION FOR TV	FT	STV	STV
153	YTV		Y POSITION FOR TV	FT	STV	STV
154	HTV		HEIGHT FOR TV	FT	STV	STV
155	XDTV		X VELOCITY FOR TV	FT/SEC	STV	
156	YDTV		Y VELOCITY FOR TV	FT/SEC	STV	
157	HDTV		H VELOCITY FOR TV	FT/SEC	STV	
158	XDOTH		RATE OF CHANGE OF A/C'S X POSN	FT/SEC	STV	DER'TIVE STV
159	XIHX		CONTROL VAR FOR X INTEGRATION		STV	DER'TIVE
160	XIHY		CONTROL VAR FOR Y INTEGRATION		STV	DER'TIVE
161	SXPL		NEAR BELT JOIN(POS)AND LIMIT	FT	STV	STV
162	SXMIN		NEAR BELT JOIN(NEG)AND LIMIT	FT	STV	STV
163	SXPLUS		CYCLE LIMIT	FT	STV	STV

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 15

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
164	SXMINUS		CYCLE LIMIT	FT	STV	STV
165	SYPL		LIMIT FOR SIDEWAYS(POSITIVE)	FT	STV	STV
166	SYMIN		LIMIT FOR SIDEWAYS(NEGATIVE)	FT	STV	STV
167	HTVLIM		CEILING AND HEIGHT LIMIT	FT	STV	STV
168	XDTMX		SLEW RATE	FT/SEC	STV	STV
169	VSHIPKT		SHIP SPEED	KT	USER	STV
170						
171	EGS		GLIDE SLOPE ERROR	DEGREES	SILS	
172	ELOC		LOCALISER ERROR	DEGREES	SILS	
173	HSLOPE		HT OF ILS BEAM AT GIVEN RANGE	FT	SILS	SINIT SILS
174	RGS		RANGE FROM G/S TRANSMITTER	FT	SILS	SILS
175	RLOC		RANGE FROM LOC TRANSMITTER	FT	SILS	SILS
176	RKINK		RANGE FROM BEAM KINK	FT	SILS	SILS
177	XGS		GLIDE SLOPE ORIGIN	FT	SILS	SILS

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 16

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
178	XLOC		LOCALISER ORIGIN (X)	FT	SILS	SILS
179	YLOC		LOCALISER ORIGIN (Y)	FT	SILS	SILS
180	SGS		GLIDE SLOPE SENSITIVITY	DEGREES	SILS	SILS
181	SLOC		LOCALISER SENSITIVITY	DEGREES	SILS	SILS
182	HKINK		HEIGHT OF BEAM KINK	FT	SILS	SILS
183	BSLOPE		SLOPE OF LOWER BEAM	DEGREES	SILS	SILS
184	USLOPE		SLOPE OF UPPER BEAM	DEGREES	SILS	SILS
185	XKINK		POSITION OF BEAM KINK	FT	SILS	SILS
186	CILS1		RADTOD/SGS		SILS	SILS
187	CILS2		RADTOD/SLOC		SILS	SILS
188	CILS3		TAN(BSLOPE)		SILS	SILS
189	CILS4		TAN(USLOPE)-CILS3		SILS	SILS
190						
191	BETADIC		INITIAL VALUE OF BETA	DEGREES	ICFILE	SINIT

SYSTEM COMMON - REAL VARIABLES 15.3.79

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 17

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
192	ALFADIC		INITIAL VALUE OF ALFA	DEGREES	ICFILE	SINIT
193	GAMDIC		INITIAL VALUE OF GAMMA	DEGREES	ICFILE	SINIT
194	VKTIC		INITIAL AIR SPEED	KNOTS	ICFILE	SINIT SWIND
195	XIC		INITIAL POSITION	FT	ICFILE	SINIT INITIAL STV
196	YIC		INITIAL POSITION	FT	ICFILE	INITIAL
197	HIC		INITIAL POSITION	FT	ICFILE	SINIT SWIND INITIAL
198	W		AIRCRAFT WEIGHT	LB	ICFILE	SINIT SACCBOD
199	XCG		X C.G. LOCATION IN AIRCRAFT		ICFILE	SINIT
200	XILSFLAG		ILSFLAG IN 'REAL' FORM		ICFILE	INITIAL
201	XISHR		ISHR IN 'REAL' FORM		ICFILE	INITIAL
202	VWKTO		DATUM WIND SPEED	KNOTS	ICFILE	SINIT

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 18

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
203	PSIWD		DATUM WIND DIRECTION	DEGREES	ICFILE	SINIT
204	PSIDIC		INITIAL VALUE OF PSI	DEGREES	ICFILE	INITIAL SINIT
205	RIC15		SPARE			
206	RIC16		SPARE			
207	RIC17		SPARE			
208	RIC18		SPARE			
209	RIC19		SPARE			
210	RIC20		SPARE			
211	VKNIC		INIT GROUND SPEED COMP (NORTH)	FT/SEC	SINIT	INITIAL SINIT
212	VKEIC		INIT GROUND SPEED COMP (EAST)	FT/SEC	SINIT	INITIAL SINIT
213	VKDIC		INIT GROUND SPEED COMP (DOWN)	FT/SEC	SINIT	INITIAL SINIT
214	PHIDIC		INITIAL VALUE OF PHI	DEGREES	SINIT	INITIAL

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 19

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
215	THETADIC		INITIAL VALUE OF THETA	DEGREES	SINIT	INITIAL
216						
217	VN		COMP. OF TRUE AIR SPEED(NORTH)	FT/SEC	SVELOC1	SVELOC1
218	VE		COMP. OF TRUE AIR SPEED(EAST)	FT/SEC	SVELOC1	SVELOC1
219	VD		COMP. OF TRUE AIR SPEED(DOWN)	FT/SEC	SVELOC1	SVELOC1
220	PDIC		INITIAL ANGULAR RATE	RADS/SEC	SINIT	INITIAL
221	QDIC		INITIAL ANGULAR RATE	RADS/SEC	SINIT	INITIAL
222	RDIC		INITIAL ANGULAR RATE	RADS/SEC	SINIT	INITIAL
223	VWDLO		WIND VELOCITY(DOWN)AT HEIGHT	FT/SEC	SWIND	SWIND
224	CPSIW		COS(PSIWD)		SINIT	SWIND SINIT
225	SPSIW		SIN(PSIWD)		SINIT	SWIND SINIT
226	VWNLO		WIND VELOCITY (NORTH)AT HEIGHT	FT/SEC	SINIT	SWIND SINIT
227	VWELO		WIND VELOCITY (EAST)AT HEIGHT	FT/SEC	SINIT	SWIND

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 20

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SINIT
228	SHRFAC		SHEAR FACTOR		WSHEAR	SINIT SWIND
229	VWNL		MEAN WIND COMP AT HEIGHT(NTH)	FT/SEC	SWIND	SWIND
230	VWEL		MEAN WIND COMP AT HEIGHT(EAST)	FT/SEC	SWIND	SWIND
231	VWDL		MEAN WIND COMP AT HEIGHT(DOWN)	FT/SEC	SWIND	SWIND
232	UTURB		COMPONENT OF TURBULENCE	FT/SEC	SWIND	SWIND
233	VTURB		COMPONENT OF TURBULENCE	FT/SEC	SWIND	SWIND
234	WTURB		COMPONENT OF TURBULENCE	FT/SEC	SWIND	SWIND
235	USIG		RMS OF TURBULENCE	FT/SEC	USER	SWIND
236	VSIG		RMS OF TURBULENCE	FT/SEC	USER	SWIND
237	WSIG		RMS OF TURBULENCE	FT/SEC	USER	SWIND
238	VKNDOT		ACCELERATION IN EARTH AXES	FT/SEC2	SACCLIN	DER'TIVE
239	VKEDOT		ACCELERATION IN EARTH AXES	FT/SEC2	SACCLIN	DER'TIVE

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 21

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
241	AXCG		SPECIFIC ACCELERATION AT CG(X)	G	SACCBOD	SACCBOD
242	AYCG		SPECIFIC ACCELERATION AT CG(Y)	G	SACCBOD	SACCBOD
243	AZCG		SPECIFIC ACCELERATION AT CG(Z)	G	SACCBOD	SACCBOD
244	AZCG1		Z ACCEL'N AT CG,REL. TO 1.0	G	SACCBOD	
245	AXACG		ABSOLUTE ACCELERATION AT CG(X)	G	SACCBOD	
246	AYACG		ABSOLUTE ACCELERATION AT CG(Y)	G	SACCBOD	
247	AZACG		ABSOLUTE ACCELERATION AT CG(Z)	G	SACCBOD	
248	AXP		SPECIFIC ACCEL'N AT PILOT (X)	G	SACCBOD	
249	AYP		SPECIFIC ACCEL'N AT PILOT (Y)	G	SACCBOD	
250	AZP		SPECIFIC ACCEL'N AT PILOT (Z)	G	SACCBOD	SACCBOD
251	AZAP		ABSOLUTE ACCEL'N AT PILOT (Z)	G	SACCBOD	
252	AY1		ACCELERATION AT SLIP BALL	G	SACCBOD	
253	AZ2		ACCELERATION AT G METER	G	SACCBOD	
254	AX3		X ACCEL'N AT (X3,Y3,Z3)	G	SACCBOD	

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 22

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
255	AY4		Y ACCEL'N AT (X4,Y4,Z4)	G	SACCBOD	
256	AZ5		Z ACCEL'N AT (X5,Y5,Z5)	G	SACCBOD	
257	X1		COORDINATES OF SLIP BALL (AY1)	FT	USER	SACCBOD
258	Y1		COORDINATES OF SLIP BALL (AY1)	FT	USER	SACCBOD
259	Z1		COORDINATES OF SLIP BALL (AY1)	FT	USER	SACCBOD
260	X2		COORDINATES OF G METER (AZ2)	FT	USER	SACCBOD
261	Y2		COORDINATES OF G METER (AZ2)	FT	USER	SACCBOD
262	Z2		COORDINATES OF G METER (AZ2)	FT	USER	SACCBOD
263	X3		COORD. OF ACCELEROMETER (AX3)	FT	USER	SACCBOD
264	Y3		COORD. OF ACCELEROMETER (AX3)	FT	USER	SACCBOD
265	Z3		COORD. OF ACCELEROMETER (AX3)	FT	USER	SACCBOD
266	X4		COORD. OF ACCELEROMETER (AY4)	FT	USER	SACCBOD
267	Y4		COORD. OF ACCELEROMETER (AY4)	FT	USER	SACCBOD
268	Z4		COORD. OF ACCELEROMETER (AY4)	FT	USER	SACCBOD

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 23

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
269	XS		COORD. OF ACCELEROMETER (AZS)	FT	USER	SACCBOD
270	YS		COORD. OF ACCELEROMETER (AZS)	FT	USER	SACCBOD
271	ZS		COORD. OF ACCELEROMETER (AZS)	FT	USER	SACCBOD
******** NOTE MISSING ELEMENT NUMBERS *****						
278	SFRACG		INTERMITTENCY FOR TURBULENCE		USER	SWIND
279	SRDECAY		DECAY FACTOR FOR TURBULENCE		USER	SWIND
280	XPCG		XLOC OF PILOT REL TO CG		STV	STV SACCBOD
281	ZPCG		Z LOC OF PILOT REL TO CG		STV	STV SACCBOD
282	RXP		PILOT X RADIUS	FT	STV	STV
283	RHP		PILOT H RADIUS	FT	STV	STV
284	XICF		EXTRA STORE FOR XIC	FT	STV	STV
******** NOTE MISSING ELEMENT NUMBERS *****						
301	AADC(1)		ARRAY OF SCALING FACTORS		USER	SADC

SYSTEM COMMON - REAL VARIABLES 15.3.78

NUMERIC ORDER 10:45 MAR 13,'79 PAGE 24

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
******** NOTE MISSING ELEMENT NUMBERS *****						
364	AADC(64)		ARRAY OF SCALING FACTORS		USER	SADC
365	TVCON(1,1)		ARRAY OF CONSTANTS FOR TV BELT		STV	STV
******** NOTE MISSING ELEMENT NUMBERS *****						
388	TVCON(8,3)		ARRAY OF CONSTANTS FOR TV BELT		STV	STV
******** NOTE MISSING ELEMENT NUMBERS *****						
401	YADC(1)		ARRAY OF DESCALED INPUTS		SADC	USER
******** NOTE MISSING ELEMENT NUMBERS *****						
464	YADC(64)		ARRAY OF DESCALED INPUTS		SADC	USER
******** NOTE MISSING ELEMENT NUMBERS *****						
501	ADAC(1)		ARRAY OF SCALING FACTORS		USER	SDAC
******** NOTE MISSING ELEMENT NUMBERS *****						
564	ADAC(64)		ARRAY OF SCALING FACTORS		USER	SDAC

D.2 Alphabetical order

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 1

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
301	AADC(1)		ARRAY OF SCALING FACTORS		USER	SADC
364	AADC(64)		ARRAY OF SCALING FACTORS		USER	SADC
501	ADAC(1)		ARRAY OF SCALING FACTORS		USER	SDAC
564	ADAC(64)		ARRAY OF SCALING FACTORS		USER	SDAC
112	ALFAD		ANGLE OF ATTACK	DEGREES	SALFBET	
192	ALFADIC		INITIAL VALUE OF ALFA	DEGREES	ICFILE	SINIT
120	ALFADOT		RATE OF ANGLE OF ATTACK	RADS/SEC	SALFBET	
111	ALFAR		ANGLE OF ATTACK	RADIANS	SALFBET	SALFBET
85	ARATIO		ADIABATIC SPEED OF SOUND RATIO		SVELOC2	SVELOC2
254	AX3		X ACCEL'N AT (X3,Y3,Z3)	G	SACCBOD	
245	AXACG		ABSOLUTE ACCELERATION AT CG(X)	G	SACCBOD	
241	AXCG		SPECIFIC ACCELERATION AT CG(X)	G	SACCBOD	SACCBOD
248	AXP		SPECIFIC ACCEL'N AT PILOT (X)	G	SACCBOD	
252	AY1		ACCELERATION AT SLIP BALL	G	SACCBOD	

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 2

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
255	AY4		Y ACCEL'N AT (X4,Y4,Z4)	G	SACCBOD	
246	AYACG		ABSOLUTE ACCELERATION AT CG(Y)	G	SACCBOD	
242	AYCG		SPECIFIC ACCELERATION AT CG(Y)	G	SACCBOD	SACCBOD
249	AYP		SPECIFIC ACCEL'N AT PILOT (Y)	G	SACCBOD	
253	AZ2		ACCELERATION AT G METER	G	SACCBOD	
256	AZ5		Z ACCEL'N AT (X5,Y5,Z5)	G	SACCBOD	
247	AZACG		ABSOLUTE ACCELERATION AT CG(Z)	G	SACCBOD	
251	AZAP		ABSOLUTE ACCEL'N AT PILOT (Z)	G	SACCBOD	
243	AZCG		SPECIFIC ACCELERATION AT CG(Z)	G	SACCBOD	SACCBOD
244	AZCG1		Z ACCEL'N AT CG,REL. TO 1.0	G	SACCBOD	
250	AZP		SPECIFIC ACCEL'N AT PILOT (Z)	G	SACCBOD	SACCBOD
114	BETAD		ANGLE OF SIDESLIP	DEGREES	SALFBET	
191	BETADIC		INITIAL VALUE OF BETA	DEGREES	ICFILE	SINIT
121	BETADOT		RATE OF ANGLE OF SIDESLIP	RADS/SEC	SALFBET	

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER　　　10:45 MAR 13,'79　　　PAGE 3

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
113	BETAR		ANGLE OF SIDESLIP	RADIANS	SALFBET	SALFBET
183	BSLOPE		SLOPE OF LOWER BEAM	DEGREES	SILS	SILS
117	CALFA		COS OF ALFA		SALFBET	
119	CBETA		COS OF BETA		SALFBET	
35	C11		INERTIA COEFFICIENT		SINIT	SACCROT
36	C12		INERTIA COEFFICIENT		SINIT	SACCROT
37	C13		INERTIA COEFFICIENT		SINIT	SACCROT
38	C14		INERTIA COEFFICIENT		SINIT	SACCROT
39	C15		INERTIA COEFFICIENT		SINIT	SACCROT SINIT
40	C16		INERTIA COEFFICIENT		SINIT	SACCROT
41	C17		INERTIA COEFFICIENT		SINIT	SACCROT
42	C18		INERTIA COEFFICIENT		SINIT	SACCROT
43	C19		INERTIA COEFFICIENT		SINIT	SACCROT
44	C110		INERTIA COEFFICIENT		SINIT	SACCROT

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER　　　10:45 MAR 13,'79　　　PAGE 4

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
186	CILS1		RADTOD/SGB		SILS	SILS
187	CILS2		RADTOD/SLOL		SILS	SILS
188	CILS3		TAN(BSLOPE)		SILS	SILS
189	CILS4		TAN(USLOPE)-CILS3		SILS	SILS
14	CPHI		COS(PHIR)		SOCOS	SEULER SOCOS
10	CPSI		COS(PSIR)		SOCOS	STV SOCOS
224	CPSIW		COS(PSIWD)		SINIT	SWIND SINIT
107	CREF		REFERENCE CHORD	FT	USER	SVELOC2 STV
12	CTHETA		COS(THETAR)		SOCOS	STV SOCOS
142	DEGTOR		CONVERSION DEGREES TO RADIANS	CONSTANT	SINIT	SILS SINIT
91	DELT1		INTEG. STEP LENGTH NO. 1	SECS	USER	INITIAL

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 5

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
95	DELT2		INTEG. STEP LENGTH NO. 2	SECS	USER	INITIAL
78	DENR		DENSITY RATIO		SINIT	SVELOC2 SINIT
83	DRATIO		ADIABATIC DENSITY RATIO		SVELOC2	SVELOC2
101	DXCG		DIST OF C.G. AHEAD OF M.R.C.		SINIT	STV
102	DZCG		DIST OF C.G. BELOW M.R.C.		SINIT	STV
171	EGS		GLIDE SLOPE ERROR	DEGREES	SILS	
172	ELOC		LOCALISER ERROR	DEGREES	SILS	
143	FPSTKT		CONVERSION FT/SEC TO KNOTS	CONSTANT	SINIT	SVELOC2 SPATH
92	FRAMET1		FRAME TIME,LOOP 1	MSEC	USER	INITIAL SWIND
96	FRAMET2		FRAME TIME,LOOP 2	MSEC	USER	INITIAL
139	FTD		TOTAL FORCE COMP IN EARTH AXES	LB	SACCLIN	SACCLIN
138	FTE		TOTAL FORCE COMP IN EARTH AXES	LB	SACCLIN	SACCLIN
137	FTN		TOTAL FORCE COMP IN EARTH AXES	LB	SACCLIN	SACCLIN

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 6

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
134	FTX		TOTAL FORCE COMP. IN BODY AXES	LB	USER	SACCLIN SACCBOD
135	FTY		TOTAL FORCE COMP. IN BODY AXES	LB	USER	SACCLIN SACCBOD
136	FTZ		TOTAL FORCE COMP. IN BODY AXES	LB	USER	SACCLIN SACCBOD
145	G		ACCELERATION DUE TO GRAVITY	CONSTANT	SINIT	SACCLIN SINIT
193	GAMDIC		INITIAL VALUE OF GAMMA	DEGREES	ICFILE	SINIT
62	GAMMAD		FLIGHT PATH ANGLE,CLIMB	DEGREES	SPATH	
61	GAMMAR		FLIGHT PATH ANGLE,CLIMB	RADIANS	SPATH	SPATH
151	H		HEIGHT OF CG	FT	DER'TIVE	SATMOS SVELOC2 STV SILS SWIND
157	HDTV		H VELOCITY FOR TV	FT/SEC	STV	
197	HIC		INITIAL POSITION	FT	ICFILE	SINIT

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 7

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SWIND INITIAL
182	HKINK		HEIGHT OF BEAM KINK	FT	SILS	SILS
124	HLFRHOV		0.5*RHO*VT		SVELOC2	SVELOC2
173	HSLOPE		HT OF ILS BEAM AT GIVEN RANGE	FT	SILS	SINIT SILS
184	HTV		HEIGHT FOR TV	FT	STV	STV
167	HTVLIM		CEILING AND HEIGHT LIMIT	FT	STV	STV
26	P		ANG VEL,ROLL,BODY AXES	RADS/SEC	DER'TIVE	SEULER SACCROT SACCBOD
86	PALPHA		FREE STREAM STATIC PRESSURE	LB/IN 2	SVELOC2	
32	PD		ANG VEL,ROLL,BODY AXES	DEGS/SEC	SEULER	
220	PDIC		INITIAL ANGULAR RATE	RADS/SEC	SINIT	INITIAL
45	PDOT		ANGULAR ACCN. IN BODY AXES	RADS/SEC2	SACCROT	SACCBOD
8	PHID		BANK ANGLE	DEGREES	SDCOS	
214	PHIDIC		INITIAL VALUE OF PHI	DEGREES	SINIT	INITIAL

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 8

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
29	PHIDT		ATTITUDE RATE,BANK	RAD/SEC	SEULER	DER'TIVE
7	PHIR		BANK ANGLE	RADIANS	DER'TIVE	SDCOS
87	PRATIO		ADIABATIC STATIC/TOTAL PRESS		SVELOC2	
77	PRESSL		AMBIENT PRESSURE AT SEA LEVEL	LB/IN 2	SINIT	SVELOC2 SINIT
82	PRESSR		ATMOSPHERIC PRESSURE RATIO		SATMOS	SVELOC2 SINIT
4	PSID		HEADING ANGLE	DEGREES	SDCOS	
204	PSIDIC		INITIAL VALUE OF PSI	DEGREES	ICFILE	INITIAL SINIT
31	PSIDT		ATTITUDE RATE,HEADING	RAD/SEC	SEULER	STV DER'TIVE
64	PSIKD		FLIGHT PATH ANGLE,TRACK	DEGREES	SPATH	
63	PSIKR		FLIGHT PATH ANGLE,TRACK	RADIANS	SPATH	SPATH
3	PSIR		HEADING ANGLE	RADIANS	DER'TIVE	SDCOS
203	PSIWD		DATUM WIND DIRECTION	DEGREES	ICFILE	SINIT

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 9

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
27	Q		ANG VEL,PITCH,BODY AXES	RADS/SEC	DER'TIVE	SEULER SACCROT SACCBOD
33	QD		ANG VEL,PITCH,BODY AXES	DEGS/SEC	SEULER	
221	QDIC		INITIAL ANGULAR RATE	RADS/SEC	SINIT	INITIAL
46	QDOT		ANGULAR ACCN.IN BODY AXES	RADS/SEC2	SACCROT	SACCBOD
125	QDYN		DYNAMIC PRESSURE	LB/FT2	SVELOC2	SVELOC2
126	QDYNS		QDYN*SWREF	LB	SVELOC2	SVELOC2
129	QSB2IV		HLFRHOV*SW2		SVELOC2	
127	QSCREF		QDYNS*CREF	LB-FT	SVELOC2	
128	QSSPAN		QDYNS*SPAN	LB-FT	SVELOC2	
28	R		ANG VEL,YAW,BODY AXES	RADS/SEC	DER'TIVE	SEULER SACCROT SACCBOD
141	RADTOD		CONVERSION RADIANS TO DEGREES	CONSTANT	SINIT	SOCOS SPATH SALFBET

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 10

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SEULER SILS SINIT
34	RD		ANG VEL,YAW,BODY AXES	DEGS/SEC	SEULER	
222	RDIC		INITIAL ANGULAR RATE	RADS/SEC	SINIT	INITIAL
47	RDOT		ANGULAR ACCN. IN BODY AXES	RADS/SEC2	SACCROT	SACCBOD
146	RG		RECIPROCAL OF G		SINIT	SACCBOD
174	RGS		RANGE FROM G/S TRANSMITTER	FT	SILS	SILS
72	RHO		AIR DENSITY	SLUG/FT3	SVELOC2	SVELOC2
79	RHOSL		DENSITY AT SEA LEVEL	SLUG /FT3	SINIT	SVELOC2 SINIT
283	RHP		PILOT H RADIUS	FT	STV	STV
205	RIC15		SPARE			
206	RIC16		SPARE			
207	RIC17		SPARE			
208	RIC18		SPARE			

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 11

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
209	RIC19		SPARE			
210	RIC20		SPARE			
176	RKINK		RANGE FROM BEAM KINK	FT	SILS	SILS
175	RLOC		RANGE FROM LOC TRANSMITTER	FT	SILS	SILS
71	ROOTSIG		SORT(DENSITY RATIO)		SVELOC2	SINIT SVELOC2
1	RXMASS		1/XMASS		SINIT	SACCLIN
282	RXP		PILOT X RADIUS	FT	STV	STV
17	S11		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
18	S12		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
19	S13		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN SACCBOD
20	S21		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 12

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
21	S22		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
22	S23		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN SACCBOD
23	S31		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
24	S32		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN
25	S33		DIRECTION COSINE		SDCOS	SVELOC1 SACCLIN SACCBOD
116	SALFA		SINE OF ALFA		SALFBET	
108	SB2		SWREF*SPAN*SPAN	FT4	SINIT	SVELOC2
118	SBETA		SINE OF BETA		SALFBET	
15	SECTHT		SEC(THETAR)		SDCOS	SEULER SDCOS
278	SFRACG		INTERMITTENCY FOR TURBULENCE		USER	SWIND

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 13

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
180	SGS		GLIDE SLOPE SENSITIVITY	DEGREES	SILS	SILS
228	SHRFAC		SHEAR FACTOR		WSHEAR	SINIT SWIND
181	SLOC		LOCALISER SENSITIVITY	DEGREES	SILS	SILS
103	SPAN		WING SPAN	FT	USER	SVELOC2 SINIT
13	SPHI		SIN(PHIR)		SDCOS	SEULER SDCOS
9	SPSI		SIN(PSIR)		SDCOS	STV SDCOS
225	SPSIW		SIN(PSIWD.		SINIT	SWIND SINIT
80	SPSL		SPEED OF SOUND AT SEA LEVEL	FT/SEC	SINIT	SVELOC2 SINIT
84	SPSNDR		SPEED OF SOUND RATIO		SATMOS	SVELOC2 SINIT
73	SPSOND		SPEED OF SOUND	FT/SEC	SVELOC2	SVELOC2
279	SRDECAY		DECAY FACTOR FOR TURBULENCE		USER	SWIND

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 14

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
105	STAIL		TAIL PLANE AREA	FT2	USER	
11	STHETA		SIN(THETAR)		SDCOS	STV SDCOS
106	SWREF		WING REFERENCE AREA	FT2	USER	SVELOC2 SINIT
162	SXMIN		NEAR BELT JOIN(NEG)AND LIMIT	FT	STV	STV
164	SXMINUS		CYCLE LIMIT	FT	STV	STV
161	SXPL		NEAR BELT JOIN(POS)AND LIMIT	FT	STV	STV
163	SXPLUS		CYCLE LIMIT	FT	STV	STV
166	SYMIN		LIMIT FOR SIDEWAYS(NEGATIVE)	FT	STV	STV
165	SYPL		LIMIT FOR SIDEWAYS(POSITIVE)	FT	STV	STV
88	TALPHA		FREE STREAM STATIC TEMPERATURE	DEG. K	SVELOC2	
16	TANTHT		TAN(THETAR)		SDCOS	SEULER
81	TEMPR		ATMOSPHERIC TEMP RATIO		SATMOS	SVELOC2 SINIT
76	TEMPSL		AMB TEMP AT SL(STANDARD DAY)	DEG. K	SINIT	SVELOC2

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 15

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SINIT
6	THETAD		PITCH ATTITUDE	DEGREES	SDCOS	
216	THETADIC		INITIAL VALUE OF THETA	DEGREES	SINIT	INITIAL
5	THETAR		PITCH ATTITUDE	RADIANS	DER'TIVE	SDCOS
30	THETDT		ATTITUDE RATE,PITCH	RAD/SEC	SEULER	STV DER'TIVE
94	TIME		TIME	SEC	DER'TIVE	
89	TRATIO		ADIABATIC STATIC/TOTAL TEMP		SVELOC2	SVELOC2
366	TVCON(1,1)		ARRAY OF CONSTANTS FOR TV BELT		STV	STV
388	TVCON(8,3)		ARRAY OF CONSTANTS FOR TV BELT		STV	STV
58	UB		VEL COMP REL TO AIR,BODY AXES	FT/SEC	SVELOC1	SVELOC2 SALFBET
235	USIG		RMS OF TURBULENCE	FT/SEC	USER	SWIND
184	USLOPE		SLOPE OF UPPER BEAM	DEGREES	SILS	SILS
232	UTURB		COMPONENT OF TURBULENCE	FT/SEC	SWIND	SWIND
59	VB		VEL COMP REL TO AIR,BODY AXES	FT/SEC	SVELOC1	SVELOC2

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 16

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SALFBET
219	VD		COMP. OF TRUE AIR SPEED(DOWN)	FT/SEC	SVELOC1	SVELOC1
218	VE		COMP. OF TRUE AIR SPEED(EAST)	FT/SEC	SVELOC1	SVELOC1
69	VEAS		EQUIVALENT AIR SPEED	FT/SEC	SVELOC2	SVELOC2
70	VEASKT		EQUIVALENT AIR SPEED	KNOTS	SVELOC2	
65	VK		VELOCITY RELATIVE TO GROUND	FT/SEC	SPATH	SINIT SPATH
64	VKD		VELOCITY REL TO GROUND,DOWN	FT/SEC	DER'TIVE	SVELOC1 SPATH STV
240	VKDDOT		ACCELERATION IN EARTH AXES	FT/SEC2	SACCLIN	DER'TIVE
213	VKDIC		INIT GROUND SPEED COMP (DOWN)	FT/SEC	SINIT	INITIAL SINIT
63	VKE		VELOCITY REL TO GROUND,EAST	FT/SEC	DER'TIVE	SVELOC1 SPATH STV
239	VKEDOT		ACCELERATION IN EARTH AXES	FT/SEC2	SACCLIN	DER'TIVE
212	VKEIC		INIT GROUND SPEED COMP (EAST)	FT/SEC	SINIT	INITIAL

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 17

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
						SINIT
66	VKKT		VELOCITY RELATIVE TO GROUND	KNOTS	SPATH	
52	VKN		VELOCITY REL TO GROUND,NORTH	FT/SEC	DER'TIVE	SVELOC1 SPATH STV
238	VKNDOT		ACCELERATION IN EARTH AXES	FT/SEC2	SACCLIN	DER'TIVE
211	VKNIC		INIT GROUND SPEED COMP (NORTH)	FT/SEC	SINIT	INITIAL SINIT
194	VKTIC		INITIAL AIR SPEED	KNOTS	ICFILE	SINIT SWIND
217	VN		COMP. OF TRUE AIR SPEED(NORTH)	FT/SEC	SVELOC1	SVELOC1
169	VSHIPKT		SHIP SPEED	KT	USER	STV
236	VSIG		RMS OF TURBULENCE	FT/SEC	USER	SWIND
67	VT		TOTAL AIR SPEED	FT/SEC	SVELOC2	SINIT SVELOC2
68	VTKT		TOTAL AIR SPEED	KNOTS	SVELOC2	
233	VTURB		COMPONENT OF TURBULENCE	FT/SEC	SWIND	SWIND

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 18

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
57	VWD		VELOCITY OF WIND,DOWN	FT/SEC	SWIND	SVELOC1
231	VWDL		MEAN WIND COMP AT HEIGHT(DOWN)	FT/SEC	SWIND	SWIND
223	VWDLO		WIND VELOCITY(DOWN)AT HEIGHT	FT/SEC	SWIND	SWIND
56	VWE		VELOCITY OF WIND,EAST	FT/SEC	SWIND	SVELOC1
230	VWEL		MEAN WIND COMP AT HEIGHT(EAST)	FT/SEC	SWIND	SWIND
227	VWELO		WIND VELOCITY (EAST)AT HEIGHT	FT/SEC	SINIT	SWIND SINIT
202	VWKTO		DATUM WIND SPEED	KNOTS	ICFILE	SINIT
55	VWN		VELOCITY OF WIND,NORTH	FT/SEC	SWIND	SVELOC1
229	VWNL		MEAN WIND COMP AT HEIGHT(NTH)	FT/SEC	SWIND	SWIND
226	VWNLO		WIND VELOCITY (NORTH)AT HEIGHT	FT/SEC	SINIT	SWIND SINIT
198	W		AIRCRAFT WEIGHT	LB	ICFILE	SINIT SACCBOD
60	WB		VEL COMP REL TO AIR,BODY AXES	FT/SEC	SVELOC1	SVELOC2 SALFBET

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 19

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
237	WSIG		RMS OF TURBULENCE	FT/SEC	USER	SWIND
234	WTURB		COMPONENT OF TURBULENCE	FT/SEC	SWIND	SWIND
149	X		X POSITION OF C.G.	FT	DER'TIVE	STV SILS
257	X1		COORDINATES OF SLIP BALL (AY1)	FT	USER	SACCBOD
260	X2		COORDINATES OF G METER (AZ2)	FT	USER	SACCBOD
263	X3		COORD. OF ACCELEROMETER (AX3)	FT	USER	SACCBOD
266	X4		COORD. OF ACCELEROMETER (AY4)	FT	USER	SACCBOD
269	X5		COORD. OF ACCELEROMETER (AZ5)	FT	USER	SACCBOD
199	XCG		X C.G. LOCATION IN AIRCRAFT		ICFILE	SINIT
99	XCGREF		X LOCATION OF REF PT		USER	SINIT
158	XDOTN		RATE OF CHANGE OF A/C'S X POSN	FT/SEC	STV	DER'TIVE STV
168	XDOTMX		SLEW RATE	FT/SEC	STV	STV
165	XDOTV		X VELOCITY FOR TV	FT/SEC	STV	

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 20

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
177	XGO		GLIDE SLOPE ORIGIN	FT	SILS	SILS
195	XIC		INITIAL POSITION	FT	ICFILE	SINIT INITIAL STV
284	XICF		EXTRA STORE FOR XIC	FT	STV	STV
159	XINX		CONTROL VAR FOR X INTEGRATION		STV	DER'TIVE
160	XINY		CONTROL VAR FOR Y INTEGRATION		STV	DER'TIVE
200	XILSFLAG		ILSFLAG IN 'REAL' FORM		ICFILE	INITIAL
201	XISHR		ISHR IN 'REAL' FORM		ICFILE	INITIAL
48	XIX		MOMENT OF INERTIA,ROLL	SLUG FT2	USER	SINIT
49	XIY		MOMENT OF INERTIA,PITCH	SLUG FT2	USER	SINIT
50	XIZ		MOMENT OF INERTIA,YAW	SLUG FT2	USER	SINIT
51	XIZX		MOMENT OF INERTIA,PRODUCT	SLUG FT2	USER	SINIT
144	XK2FPS		CONVERSION KNOTS TO FT/SEC	CONSTANT	SINIT	SWIND STV SINIT

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 21

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
188	XKINK		POSITION OF BEAM KINK	FT	SILS	SILS
131	XLLTOT		TOTAL MOMENT,ROLL,BODY AXES	LB-FT	USER	SACCROT
178	XLOC		LOCALISER ORIGIN (X)	FT	SILS	SILS
104	XLTAIL		TAIL ARM FROM M.H.C.	FT	USER	
78	XMACH		MACH NUMBER		SVELOC2	SVELOC2
2	XMASS		AIRCRAFT MASS	SLUG	SINIT	SINIT
132	XMMTOT		TOTAL MOMENT,PITCH,BODY AXES	LB-FT	USER	SACCROT
133	XNNTOT		TOTAL MOMENT,YAW,BODY AXES	LB-FT	USER	SACCROT
109	XP		X LOC OF PILOT REL TO REFCG	FT	USER	STV
280	XPCG		XLOC OF PILOT REL TO CG		STV	STV SACCBOD
152	XTV		X POSITION FOR TV	FT	STV	STV
150	Y		Y POSITION OF C.G.	FT	DER'TIVE	STV SILS
258	Y1		COORDINATES OF SLIP BALL (AY1)	FT	USER	SACCBOD

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 22

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
261	Y2		COORDINATES OF G METER (AZ2)	FT	USER	SACCBOD
264	Y3		COORD. OF ACCELEROMETER (AX3)	FT	USER	SACCBOD
267	Y4		COORD. OF ACCELEROMETER (AY4)	FT	USER	SACCBOD
270	Y5		COORD. OF ACCELEROMETER (AZ5)	FT	USER	SACCBOD
401	YADC(1)		ARRAY OF UNSCALED INPUTS		SADC	USER
464	YADC(64)		ARRAY OF UNSCALED INPUTS		SADC	USER
156	YDTV		Y VELOCITY FOR TV	FT/SEC	STV	
196	YIC		INITIAL POSITION	FT	ICFILE	INITIAL
179	YLOC		LOCALISER ORIGIN (Y)	FT	SILS	SILS
153	YTV		Y POSITION FOR TV	FT	STV	STV
259	Z1		COORDINATES OF SLIP BALL (AY1)	FT	USER	SACCBOD
262	Z2		COORDINATES OF G METER (AZ2)	FT	USER	SACCBOD
265	Z3		COORD. OF ACCELEROMETER (AX3)	FT	USER	SACCBOD
268	Z4		COORD. OF ACCELEROMETER (AY4)	FT	USER	SACCBOD

SYSTEM COMMON - REAL VARIABLES 15.3.78

ALPHABETIC ORDER 10:45 MAR 13,'79 PAGE 23

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
271	ZS		COORD. OF ACCELEROMETER (AZS)	FT	USER	SACCBOD
98	ZCG		Z C.G. LOCATION		USER	SINIT
100	ZCGREF		Z LOCATION OF REF PT		USER	SINIT
110	ZP		Z LOC OF PILOT REL TO REFCG	FT	USER	STV
281	ZPCG		Z LOC OF PILOT REL TO CG		STV	STV SACCBOD

Appendix E

INDEX TO SYSTEM COMMON, INTEGER VARIABLES

E.1 Numeric order

SYSTEM COMMON - INTEGER VARIABLES 16-03-78

NUMERIC ORDER 10:50 MAR 13,'79 PAGE 1

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
1	KISA		SELECT ATMOSPHERE VARIATION		USER	SVELOC2 SINIT
2	NCPASS		COUNT OF PASSES THRO DER'TIVE		INITIAL	DER'TIVE SCOUNT
3	NIPASS		COUNT OF PASSES TO SUIT INTEG		INITIAL	DER'TIVE
4	ILSFLG		SELECTS ILS BEAM(SEE XILSFLG)		INITIAL	SILS
5	ISHR		WIND SHEAR SHAPE(SEE XISHR)		INITIAL	SWIND SINIT
6	ISYS		UNIT NO FOR SYSTEM CHANGES		USER	SYSCOM
7	NTVS		TV BELT SELECT		USER	STV
8	JJCOMP		INITIALISATION COMPLETE FLAG		SCOUNT	SILS STV SWIND
9	NUMD		NO. OF DACS TO BE SET FROM 'A'		USER	SDAC
10	NUMV		NO.OF VARIDACS BE SET FROM 'A'		USER	SDAC
11	NDAC(1)		POINTER TO VARIABLE FOR DAC 1		USER	SDAC

******** NOTE MISSING ELEMENT NUMBERS *****

SYSTEM COMMON - INTEGER VARIABLES 16-03-78

NUMERIC ORDER 10:50 MAR 13,'79 PAGE 2

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
******** NOTE MISSING ELEMENT NUMBERS *****						
74	NDAC(64)		POINTER TO VARIABLE FOR DAC 64		USER	SDAC
75	KSADC		VERSION ID - ROUTINE SADC		SADC	
76	KSDAC		VERSION ID - ROUTINE SDAC		SDAC	
77	KDSCRT1		VERSION ID - ROUTINE DSCRT1		DSCRT1	
78	KDSCRT2		VERSION ID - ROUTINE DSCRT2		DSCRT2	
79	KISDSCR		VERSION ID - ROUTINE ISDSCR		ISDSCR	
80						
81	NADC(1)		ARRAY OF ADC CHANNEL NUMBERS		USER	SADC
******** NOTE MISSING ELEMENT NUMBERS *****						
144	NADC(64)		ARRAY OF ADC CHANNEL NUMBERS		USER	SADC
145						
146						
147						

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

NUMERIC ORDER 10:50 MAR 13,'79 PAGE 3

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
148	LSILS		CONTROLS EXECUTION OF SILS		USER	SILS
149						
150	IAD4SL		CONTROLS EXECUTION OF READSLR		USER	DSCRT1 DSCRT2
151	ISLR(1)		AD4 SENSE LINE 1		READSLR	USER
******** NOTE MISSING ELEMENT NUMBERS *****						
166	ISLR(16)		AD4 SENSE LINE 16		READSLR	USER
167	NG		SCALE FACTOR IN NGUSTS		USER	SWIND
168	LSEED		CONTROLS RANDOM NUMBER SEED		USER	SWIND
169	NRUN		CURRENT RUN NO. FROM MAILBOX		SCOUNT	
170	IAD4CL		CONTROLS EXECUTION OF SETCLR		USER	DSCRT1 DSCRT2
171	ICLR(1)		AD4 CONTROL LINE 1		USER	SETCLR
******** NOTE MISSING ELEMENT NUMBERS *****						
186	ICLR(16)		AD4 CONTROL LINE 16		USER	SETCLR

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

NUMERIC ORDER 10:50 MAR 13,'79 PAGE 4

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
187						
188						
189						
190	IDSCFL		CONTROLS EXECUTION OF READSCR		USER	DSCRT1 DSCRT2
191	IP(1)		PATCHABLE DISCRETE 1 TO SIGMA		READSCR	ISDSCR
******** NOTE MISSING ELEMENT NUMBERS *****						
222	IP(32)		PATCHABLE DISCRETE 32 TO SIGMA		READSCR	ISDSCR
223	IDS(1)		DESK SWITCH 1 TO SIGMA		READSCR	INITIAL ISDSCR
******** NOTE MISSING ELEMENT NUMBERS *****						
254	IDS(32)		DESK SWITCH 32 TO SIGMA		READSCR	ISDSCR
255	KSACCBOD		VERSION ID - ROUTINE SACCBOD		SACCBOD	
256	KSACCLIN		VERSION ID - ROUTINE SACCLIN		SACCLIN	
257	KSACCROT		VERSION ID - ROUTINE SACCROT		SACCROT	

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

NUMERIC ORDER 10:50 MAR 13,'79 PAGE 5

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
258	KSALFBET		VERSION ID - ROUTINE SALFBET		SALFBET	
259	KSCOUNT		VERSION ID - ROUTINE SCOUNT		SCOUNT	
260	KSDCOS		VERSION ID - ROUTINE SDCOS		SDCOS	
261	KSEULER		VERSION ID - ROUTINE SEULER		SEULER	
262	KSILS		VERSION ID - ROUTINE SILS		SILS	
263	KSINIT		VERSION ID - ROUTINE SINIT		SINIT	
264	KSPATH		VERSION ID - ROUTINE SPATH		SPATH	
265	KSTV		VERSION ID - ROUTINE STV		STV	
266	KSVELOC1		VERSION ID - ROUTINE SVELOC1		SVELOC1	
267	KSVELOC2		VERSION ID - ROUTINE SVELOC2		SVELOC2	
268	KSWIND		VERSION ID - ROUTINE SWIND		SWIND	
269	KSYSCOM		VERSION ID - ROUTINE SYSCOM		SYSCOM	
270	IPCOFL		CONTROLS EXECUTION OF SETDSCR		USER	DSCRT1 DSCRT2
271	ICO(1)		CHANGE OVER 1 OUT OF SIGMA		USER	SETDSCR

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

NUMERIC ORDER 10:50 MAR 13,'79 PAGE 6

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
******** NOTE MISSING ELEMENT NUMBERS *****						
302	ICO(32)		CHANGE OVER 32OUT OF SIGMA		USER	SETDSCR
303						
304						
305	L1DAC(1)		DAC SOURCE POINTERS,LOOP1		USER	SDACN
******** NOTE MISSING ELEMENT NUMBERS *****						
352	L1DAC(48)		DAC SOURCE POINTERS,LOOP1		USER	SDACN
353	L2DAC(1)		DAC SOURCE POINTERS,LOOP2		USER	SDACN
******** NOTE MISSING ELEMENT NUMBERS *****						
400	L2DAC(48)		DAC SOURCE POINTERS,LOOP2		USER	SDACN

E.2 Alphabetic order

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

ALPHABETIC ORDER 10:50 MAR 13,'79 PAGE 1

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
170	IADACL		CONTROLS EXECUTION OF SETCLR		USER	DSCRT1 DSCRT2
150	IADASL		CONTROLS EXECUTION OF READSLR		USER	DSCRT1 DSCRT2
171	ICLR(1)		ADA CONTROL LINE 1		USER	SETCLR
186	ICLR(16)		ADA CONTROL LINE 16		USER	SETCLR
271	ICO(1)		CHANGE OVER 1 OUT OF SIGMA		USER	SETDSCR
302	ICO(32)		CHANGE OVER 32OUT OF SIGMA		USER	SETDSCR
223	IDS(1)		DESK SWITCH 1 TO SIGMA		READSCR	INITIAL ISDSCR
254	IDS(32)		DESK SWITCH 32 TO SIGMA		READSCR	ISDSCR
190	IDSCFL		CONTROLS EXECUTION OF READSCR		USER	DSCRT1 DSCRT2
4	ILSFLG		SELECTS ILS BEAM(SEE XILSFLG)		INITIAL	SILS
191	IP(1)		PATCHABLE DISCRETE 1 TO SIGMA		READSCR	ISDSCR
222	IP(32)		PATCHABLE DISCRETE 32 TO SIGMA		READSCR	ISDSCR

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

ALPHABETIC ORDER 10:50 MAR 13,'79 PAGE 2

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
270	IPCOFL		CONTROLS EXECUTION OF SETDSCR		USER	DSCRT1 DSCRT2
5	ISHR		WIND SHEAR SHAPE(SEE XISHR)		INITIAL	SWIND SINIT
151	ISLR(1)		ADA SENSE LINE 1		READSLR	USER
166	ISLR(16)		ADA SENSE LINE 16		READSLR	USER
6	ISYS		UNIT NO FOR SYSTEM CHANGES		USER	SYSCOM
8	JJCOMP		INITIALISATION COMPLETE FLAG		SCOUNT	SILS STV SWIND
77	KDSCRT1		VERSION ID - ROUTINE DSCRT1		DSCRT1	
78	KDSCRT2		VERSION ID - ROUTINE DSCRT2		DSCRT2	
1	KISA		SELECT ATMOSPHERE VARIATION		USER	SVELOC2 SINIT
79	KISDSCR		VERSION ID - ROUTINE ISDSCR		ISDSCR	
255	KSACCBOD		VERSION ID - ROUTINE SACCBOD		SACCBOD	
256	KSACCLIN		VERSION ID - ROUTINE SACCLIN		SACCLIN	

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

ALPHABETIC ORDER 10:50 MAR 13,'79 PAGE 3

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
257	KSACCROT		VERSION ID - ROUTINE SACCROT		SACCROT	
75	KSADC		VERSION ID - ROUTINE SADC		SADC	
258	KSALFBET		VERSION ID - ROUTINE SALFBET		SALFBET	
259	KSCOUNT		VERSION ID - ROUTINE SCOUNT		SCOUNT	
76	KSDAC		VERSION ID - ROUTINE SDAC		SDAC	
260	KSDCOS		VERSION ID - ROUTINE SDCOS		SDCOS	
261	KSEULER		VERSION ID - ROUTINE SEULER		SEULER	
262	KSILS		VERSION ID - ROUTINE SILS		SILS	
263	KSINIT		VERSION ID - ROUTINE SINIT		SINIT	
264	KSPATH		VERSION ID - ROUTINE SPATH		SPATH	
265	KSTV		VERSION ID - ROUTINE STV		STV	
266	KSVELOC1		VERSION ID - ROUTINE SVELOC1		SVELOC1	
267	KSVELOC2		VERSION ID - ROUTINE SVELOC2		SVELOC2	
268	KSWIND		VERSION ID - ROUTINE SWIND		SWIND	

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

ALPHABETIC ORDER 10:50 MAR 13,'79 PAGE 4

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
269	KSYSCOM		VERSION ID - ROUTINE SYSCOM		SYSCOM	
305	L1DAC(1)		DAC SOURCE POINTERS,LOOP1		USER	SDACN
352	L1DAC(48)		DAC SOURCE POINTERS,LOOP1		USER	SDACN
353	L2DAC(1)		DAC SOURCE POINTERS,LOOP2		USER	SDACN
400	L2DAC(48)		DAC SOURCE POINTERS,LOOP2		USER	SDACN
168	LSEED		CONTROLS RANDOM NUMBER SEED		USER	SWIND
148	LSILS		CONTROLS EXECUTION OF SILS		USER	SILS
81	NADC(1)		ARRAY OF ADC CHANNEL NUMBERS		USER	SADC
144	NADC(64)		ARRAY OF ADC CHANNEL NUMBERS		USER	SADC
2	NCPASS		COUNT OF PASSES THRO DER'TIVE		INITIAL	DER'TIVE SCOUNT
11	NDAC(1)		POINTER TO VARIABLE FOR DAC 1		USER	SDAC
74	NDAC(64)		POINTER TO VARIABLE FOR DAC 64		USER	SDAC
167	NG		SCALE FACTOR IN NGUSTS		USER	SWIND
3	NIPASS		COUNT OF PASSES TO SUIT INTEG		INITIAL	DER'TIVE

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

ALPHABETIC ORDER

10:50 MAR 13,'79

PAGE 5

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	CALCULATED IN	USED IN
169	NRUN		CURRENT RUN NO. FROM MAILBOX		SCOUNT	
7	NTVB		TV BELT SELECT		USER	STV
9	NUMD		NO. OF DACS TO BE SET FROM 'A'		USER	SDAC
10	NUMV		NO.OF VARIDACS BE SET FROM 'A'		USER	SDAC

Appendix F

DESCRIPTION OF PROGRAM COMMLIST

F.1 Introduction

The program COMMLIST produces an index to the variable names used in a large labelled COMMON area. A data record is prepared on cards for each variable name, giving a brief definition of the variable, its units and where it is calculated and used. A collection of such cards (which may be held on a disc file) may be sorted according to a number of criteria, of which the most important are alphabetic order of name and numeric order of element number.

For simulator purposes, two large COMMON areas are in current use, one defining system variables which are the same from simulation to simulation, *eg* height, speed and the other defining user variables which are entirely specific to the particular aircraft being simulated.

F.2 Input data

F.2.1 Main data record

Column	Purpose	Length
1 - 4	Element number	4 digits
5 - 16	Variable name	12 characters
18 - 47	Description	30 characters
49 - 58	Units	10 characters
60 - 67	Name of routine in which variable is calculated	8 characters
69 - 76	Name of routine in which variable is used	8 characters
78	Continuation indicator	1 character
	blank = no continuation	
	C = continuation .	

In general, no field delimiters are required. However, it may help human reading of data cards to insert a comma between fields. This is allowed, but is optional. The element number should be right justified in its field. A minimum of 2 data cards must be input.

F.2.2 Continuation cards

If the continuation indicator is set (C in column 78), the next card is read for additional 'used in' subroutine names. The format of this continuation card is up to 8 fields of 8 characters, separated by commas, a blank field signifying the end of the list of subroutine names and a field containing Cbbbbbbb signifying continuation on the next card.

008

F.3 Complete job deck

F.3.1 Original data on cards

```
!JOB SIM,BARRY LIST COMMON VARIABLES
!ATTEND
!COMMLIST
     Title card
     Selection card
     Data cards
!EOD
!FIN
```

F.3.2 Original data on disc file

It is assumed data cards have been copied to a file by :COPY

```
!JOB SIM,BARRY LIST COMMON VARIABLES
!ATTEND
!ASSIGN (F:100,DC,RCOMMON)
!COMMLIST
     Title card
     Selection card
     Additional data cards (optional)
!EOD
!FIN
```

F.3.3 Remarks

Note, in example section F.3.2 above, that although the main input file is taken to be a disc file, data can still be read from the card reader in the same run. This enables additional data cards to be added to and merged with the main file, thus avoiding the need to recreate the main file for every small change in data.

Note also that if original data is on cards only, as in example section F.3.1, device 100 does *not* need to be assigned to the card reader.

!EOD is essential.

F.3.4 Title card

The 80 characters on this card are printed out as a heading on each page of the output listing exactly as they appear on the card. This enables the user to identify the nature of the information being listed. If the contents of the title card are centred on the card, they will be centred on the listing page.

F.3.5 Selection of listing options

Options are specified on a selection card, containing up to 20 words of 4 characters. The options are

Abbb	Alphabetic order of variable name
Dbbb	Data listed as stored
Nbbb	Numeric order according to element number
CbbbSSSSSSSS	Lists all variables calculated in subroutine SSSSSSSS
UbbbSSSSSSSS	Lists all variables used in subroutine SSSSSSSS
Xbbb	Read next card for additional instructions.

The blanks (b = blank) are essential. There are no field separators.

As many options as desired may be specified. If they will not fit on one selection card, continuation is indicated by the X option.

F.4 Output listings

The output listing reproduces the input information spaced out for better legibility. An extra column labelled 'quantity' is left blank for the insertion of mathematical symbols associated with the variable.

For the alphabetical option, all blank records (*ie* those with only an element number) are ignored and not printed.

For the numeric option, a warning message is output when an unused element number is encountered.

For options C and U the 'calculated in' or 'used in' columns are omitted, as appropriate.

In the early stages of building a COMMON list, it is worth including data cards containing only an element number. These will be listed by the numeric option and enable details of new variables to be written in on the listing.

Appendices D, E, G and H were produced by the COMMLIST program.

F.5 Arrays

It is recommended that the first and last elements of an array are included as data items, in order to make it clear in the listings which items in the complete list are genuinely unused. COMMLIST will handle arrays of more than one dimension, provided that an element, *eg* X(12,10) can be fitted into the allowable 12 characters. If this recommendation is adopted, the user will find the warning message 'NOTE MISSING ELEMENT NUMBERS' interspersed between the first and last elements in the numeric order listing.

008

F.6 Operational considerations

F.6.1 Devices used

DCB	Device	Purpose
100	defaults to card reader	data file
101	console TTY	opening message (!!COMMLIST)
105	card reader	title card
105	card reader	selection (control) cards
105	card reader	extra data
108	line printer	all listings

F.6.2 Messages

F.6.2.1 !!COMMLIST

This message on the console teletype announces the start of execution of COMMLIST.

F.6.2.2 *** NOTE MISSING ELEMENT NUMBERS ***

This is a warning message to draw attention to gaps in the sequence of element numbers.

F.6.2.3 COMMLIST ABORTED - TOO MANY INPUT RECORDS

This message is output on the teletype when too many input records have been read. It is repeated on device 108, supplemented by the last record read.

F.7 Limits on data

This program uses many (23) arrays to hold the input information. These arrays are dimensioned at present to cater for 500 input data cards. However, since the DO loops are in the range 1 to 1000, the number of input data cards could be increased to 1000 by altering the DIMENSION statements in the program and by increasing JMAX, the limit on the number of records read. Any increase will greatly enlarge the core store needed to run the program, so should not be made unless really necessary. This present limit of 500 applies, of course, to individual variable names, so the program will cope with a user array of 2000 or more locations if many constituents are themselves large arrays, for which it is adequate to include only the first and last elements, as already discussed.

F.8 Location of system COMMON

The raw data for system COMMON variables are held in files on the DC area of the disc. Real variables are in file DC, RCOMMON and integer variables in file DC, ICOMMON. An index listing may therefore be made at any time using the technique of section F.3.2, with the job card

!ASSIGN (F:100,DC,RCOMMON)

or

!ASSIGN (F:100,DC,ICOMMON)

included as required.

Appendix G

INDEX TO SYSTEM DATA TO BE PROVIDED BY THE USER, REAL VARIABLES

SYSTEM COMMON - REAL VARIABLES 15.3.78

CALCULATED IN USER 10:45 MAR 13,'79 PAGE 1

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	USED IN
48	XIX		MOMENT OF INERTIA,ROLL	SLUG FT2	SINIT
49	XIY		MOMENT OF INERTIA,PITCH	SLUG FT2	SINIT
50	XIZ		MOMENT OF INERTIA,YAW	SLUG FT2	SINIT
51	XIZX		MOMENT OF INERTIA,PRODUCT	SLUG FT2	SINIT
91	DELT1		INTEG. STEP LENGTH NO. 1	SECS	INITIAL
92	FRAMET1		FRAME TIME,LOOP 1	MSEC	INITIAL SWIND
95	DELT2		INTEG. STEP LENGTH NO. 2	SECS	INITIAL
96	FRAMET2		FRAME TIME,LOOP 2	MSEC	INITIAL
98	ZCG		Z C.G. LOCATION		SINIT
99	XCGREF		X LOCATION OF REF PT		SINIT
100	ZCGREF		Z LOCATION OF REF PT		SINIT
103	SPAN		WING SPAN	FT	SVELOC2 SINIT
104	XLTAIL		TAIL ARM FROM M.R.C.	FT	

SYSTEM COMMON - REAL VARIABLES 15.3.78

CALCULATED IN USER 10:45 MAR 13,'79 PAGE 2

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	USED IN
105	STAIL		TAIL PLANE AREA	FT2	
106	SWREF		WING REFERENCE AREA	FT2	SVELOC2 SINIT
107	CREF		REFERENCE CHORD	FT	SVELOC2 STV
109	XP		X LOC OF PILOT REL TO REFCG	FT	STV
110	ZP		Z LOC OF PILOT REL TO REFCG	FT	STV
131	XLLTOT		TOTAL MOMENT,ROLL,BODY AXES	LB-FT	SACCROT
132	XMMTOT		TOTAL MOMENT,PITCH,BODY AXES	LB-FT	SACCROT
133	XNNTOT		TOTAL MOMENT,YAW,BODY AXES	LB-FT	SACCROT
134	FTX		TOTAL FORCE COMP. IN BODY AXES	LB	SACCLIN SACCBOD
135	FTY		TOTAL FORCE COMP. IN BODY AXES	LB	SACCLIN SACCBOD
136	FTZ		TOTAL FORCE COMP. IN BODY AXES	LB	SACCLIN SACCBOD
169	VSHIPKT		SHIP SPEED	KT	STV

SYSTEM COMMON - REAL VARIABLES 15.3.78

CALCULATED IN USER 10:45 MAR 13,'79 PAGE 3

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	USED IN
235	USIG		RMS OF TURBULENCE	FT/SEC	SWIND
236	VSIG		RMS OF TURBULENCE	FT/SEC	SWIND
237	WSIG		RMS OF TURBULENCE	FT/SEC	SWIND
257	X1		COORDINATES OF SLIP BALL (AY1)	FT	SACCBOD
258	Y1		COORDINATES OF SLIP BALL (AY1)	FT	SACCBOD
259	Z1		COORDINATES OF SLIP BALL (AY1)	FT	SACCBOD
260	X2		COORDINATES OF G METER (AZ2)	FT	SACCBOD
261	Y2		COORDINATES OF G METER (AZ2)	FT	SACCBOD
262	Z2		COORDINATES OF G METER (AZ2)	FT	SACCBOD
263	X3		COORD. OF ACCELEROMETER (AX3)	FT	SACCBOD
264	Y3		COORD. OF ACCELEROMETER (AX3)	FT	SACCBOD
265	Z3		COORD. OF ACCELEROMETER (AX3)	FT	SACCBOD
266	X4		COORD. OF ACCELEROMETER (AY4)	FT	SACCBOD
267	Y4		COORD. OF ACCELEROMETER (AY4)	FT	SACCBOD

SYSTEM COMMON - REAL VARIABLES 15.3.78

CALCULATED IN USER 10:45 MAR 13,'79 PAGE 4

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	USED IN
268	Z4		COORD. OF ACCELEROMETER (AY4)	FT	SACCBOD
269	X5		COORD. OF ACCELEROMETER (AZ5)	FT	SACCBOD
270	Y5		COORD. OF ACCELEROMETER (AZ5)	FT	SACCBOD
271	Z5		COORD. OF ACCELEROMETER (AZ5)	FT	SACCBOD
278	SFRACG		INTERMITTENCY FOR TURBULENCE		SWIND
279	SRDECAY		DECAY FACTOR FOR TURBULENCE		SWIND
301	AADC(1)		ARRAY OF SCALING FACTORS		SADC
364	AADC(64)		ARRAY OF SCALING FACTORS		SADC
501	ADAC(1)		ARRAY OF SCALING FACTORS		SDAC
564	ADAC(64)		ARRAY OF SCALING FACTORS		SDAC

STOP 0
1FIN

Appendix H

INDEX TO SYSTEM DATA TO BE PROVIDED BY THE USER, INTEGER VARIABLES

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

CALCULATED IN USER — 10:50 MAR 13,'79 — PAGE 1

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	USED IN
1	KISA		SELECT ATMOSPHERE VARIATION		SVELOC2 SINIT
6	ISYS		UNIT NO FOR SYSTEM CHANGES		SYSCOM
7	NTVB		TV BELT SELECT		STV
9	NUMD		NO. OF DACS TO BE SET FROM 'A'		SDAC
10	NUMV		NO.OF VARIDACS BE SET FROM 'A'		SDAC
11	NDAC(1)		POINTER TO VARIABLE FOR DAC 1		SDAC
74	NDAC(64)		POINTER TO VARIABLE FOR DAC 64		SDAC
81	NADC(1)		ARRAY OF ADC CHANNEL NUMBERS		SADC
144	NADC(64)		ARRAY OF ADC CHANNEL NUMBERS		SADC
148	LSILS		CONTROLS EXECUTION OF SILS		SILS
150	IAD4SL		CONTROLS EXECUTION OF READSLR		DSCRT1 DSCRT2
167	NG		SCALE FACTOR IN NGUSTS		SWIND
168	LSEED		CONTROLS RANDOM NUMBER SEED		SWIND

SYSTEM COMMON - INTEGER VARIABLES 16.03.78

CALCULATED IN USER — 10:50 MAR 13,'79 — PAGE 2

ELEMENT NO.	FORTRAN NAME	QUANTITY	DESCRIPTION	UNITS	USED IN
170	IAD4CL		CONTROLS EXECUTION OF SETCLR		DSCRT1 DSCRT2
171	ICLR(1)		AD4 CONTROL LINE 1		SETCLR
186	ICLR(16)		AD4 CONTROL LINE 16		SETCLR
190	IDSCFL		CONTROLS EXECUTION OF READSCR		DSCRT1 DSCRT2
270	IPCOFL		CONTROLS EXECUTION OF SETDSCR		DSCRT1 DSCRT2
271	ICO(1)		CHANGE OVER 1 OUT OF SIGMA		SETDSCR
302	ICO(32)		CHANGE OVER 32OUT OF SIGMA		SETDSCR
305	L1DAC(1)		DAC SOURCE POINTERS,LOOP1		SDACN
352	L1DAC(48)		DAC SOURCE POINTERS,LOOP1		SDACN
353	L2DAC(1)		DAC SOURCE POINTERS,LOOP2		SDACN
400	L2DAC(48)		DAC SOURCE POINTERS,LOOP2		SDACN

STOP 0
IFIN

LIST OF SYMBOLS

a	speed of sound
a_x, a_y, a_z	accelerations along x, y, z body axes
b	reference wing span
$\bar{c}$	mean chord
c_{ref}	reference chord
CI_1 etc	inertia constants
f	shear factor
F_{TN}, F_{TE}, F_{TD}	applied forces in earth axes
F_{TX}, F_{TY}, F_{TZ}	applied forces in body axes
g	acceleration due to gravity
h	height of centre of gravity
I_x, I_y, I_z, I_{zx}	moments of inertia
L, M, N	total moments
M	Mach number
$\ell_1, \ell_2, \ell_3, m_1, n_1$, etc	direction cosines
m	aircraft mass
P	atmospheric pressure
p, q, r	angular velocity components
R_{hp}	vertical distance of pilot's eye above centre of gravity
R_{xp}	horizontal distance of pilot's eye from centre of gravity
S_{ij}	direction cosines ($S_{11} = \ell_1$ etc)
S	transformation matrix
S_W	reference wing area
T	atmospheric temperature
u, v, w	velocity components
V	equivalent airspeed
V_K	ground speed
V_T	true airspeed
V_N, V_E, V_D	components of airspeed in earth axes (north, east, down)
V_{KN}, V_{KE}, V_{KD}	components of velocity relative to earth
V_{WN}, V_{WE}, V_{WD}	components of total wind velocity relative to earth
$V_{WNLØ}, V_{WELØ}, V_{WDLØ}$	components of datum mean wind speed in earth axes
$V_{WKTØ}$	datum mean wind speed
$V_{WNL}, V_{WEL}, V_{WDL}$	components of mean wind speed at height
W	aircraft weight
X	force along x-axis
x, y, z	positional coordinates

008

LIST OF SYMBOLS (concluded)

x_p, z_p	coordinates of pilot's eye point, relative to reference centre of gravity
x_{pcg}, z_{pcg}	coordinates of pilot's eye point, relative to actual centre of gravity
$\Delta x_{cg}, \Delta z_{cg}$	centre of gravity displacements from reference position
α	angle of attack
β	angle of sideslip
γ	climb angle
θ	pitch attitude
ρ	air density
σ	atmospheric density ratio
ϕ	bank angle
χ	track angle
ψ	heading angle
ψ_W	wind direction

Suffices

a	absolute
B	body axes
cg	at, or of, the centre of gravity
G	gusts
L	at arbitrary location
p	pilot
r	ratio
SIG	rms
SL	sea level
T	turbulence
TV	television (for outside world display)
α	ambient

A dot over a variable denotes differentiation with respect to time.

REFERENCES

No.	Author	Title, etc
1	B.N. Tomlinson	Hybrid computing for real-time flight simulation. UKSC 'Conference on Computer Simulation', Bowness-on-Windermere, May 1975. RAE Technical Memorandum FS 73 (1976) Published in 'Simulation Councils Proceedings Series', Vol 6, No.2, December 1976, by The Society for Computer Simulation
2	W.J. Duncan	The principles of the control and stability of aircraft. Cambridge University Press (1952)
3	B. Etkin	Dynamics of flight. John Wiley, New York (1959)
4	H.R. Hopkin	A scheme of notation and nomenclature for aircraft dynamics and associated aerodynamics. RAE Technical Report 66200 (Parts 1-5) (1966) ARC R & M No.3562, Parts 1-5 (1970)
5	R.M. Howe	Coordinate systems for solving the three-dimensional flight equations. WADC Technical Note 55-747, June 1956
6	L.E. Fogarty R.M. Howe	Computer mechanization of six degree-of-freedom flight equations. NASA CR 1344, May 1969
7	R.E. McFarland	A standard kinematic model for flight simulation at NASA-Ames. NASA CR 2497, January 1975
8		The equations of motion of a rigid aircraft. Engineering Sciences Data Item No.67003. Royal Aeronautical Society, September 1966
9		Geometric and kinematic relationships for various axis systems. Engineering Sciences Data Item No.67036 Royal Aeronautical Society, September 1967
10		Direction and incidence angles. Engineering Sciences Data Item No.67037 Royal Aeronautical Society, October 1967

REFERENCES (concluded)

No.	Author	Title, etc
11		Atmospheric data for performance calculations. Engineering Sciences Data Item No.68046 Royal Aeronautical Society, December 1968
12	B.N. Tomlinson	Developments in the simulation of atmospheric turbulence. AGARD Symposium 'Flight Simulation/Guidance Systems Simulation'. The Hague, Netherlands, October 1975. AGARD Conference Proceedings CP-198, Paper 19 RAE Technical Memorandum FS 46 (1975)
13	J.G. Jones	Modelling of gusts and wind shear for aircraft assessment and certification. Presented at CAARC Symposium on Operational Problems, India. Government of India Aeronautics R & D Board, ARDB-Proc-701 (1976)
14		Xerox SL1 Reference Manual Document 90 16 76B Xerox Data Systems, February 1972
15		Real Time SL1 Reference Manual Version A Rank Xerox Data Systems, February 1974
16		SL1 Technical Manual Document 70 29 46 Xerox Data Systems
17	J.L.M. Robertson	Data Logging, Parameter Variation and Inspection Systems Operations Manual. RAE Internal Document S-01-DLRV, November 1975
18	H.E. Bean	Aircraft simulation system. Honeywell Aerospace Report T-331, April 1970
19	B. Etkin	Dynamics of atmospheric flight. John Wiley, New York (1972)

)08

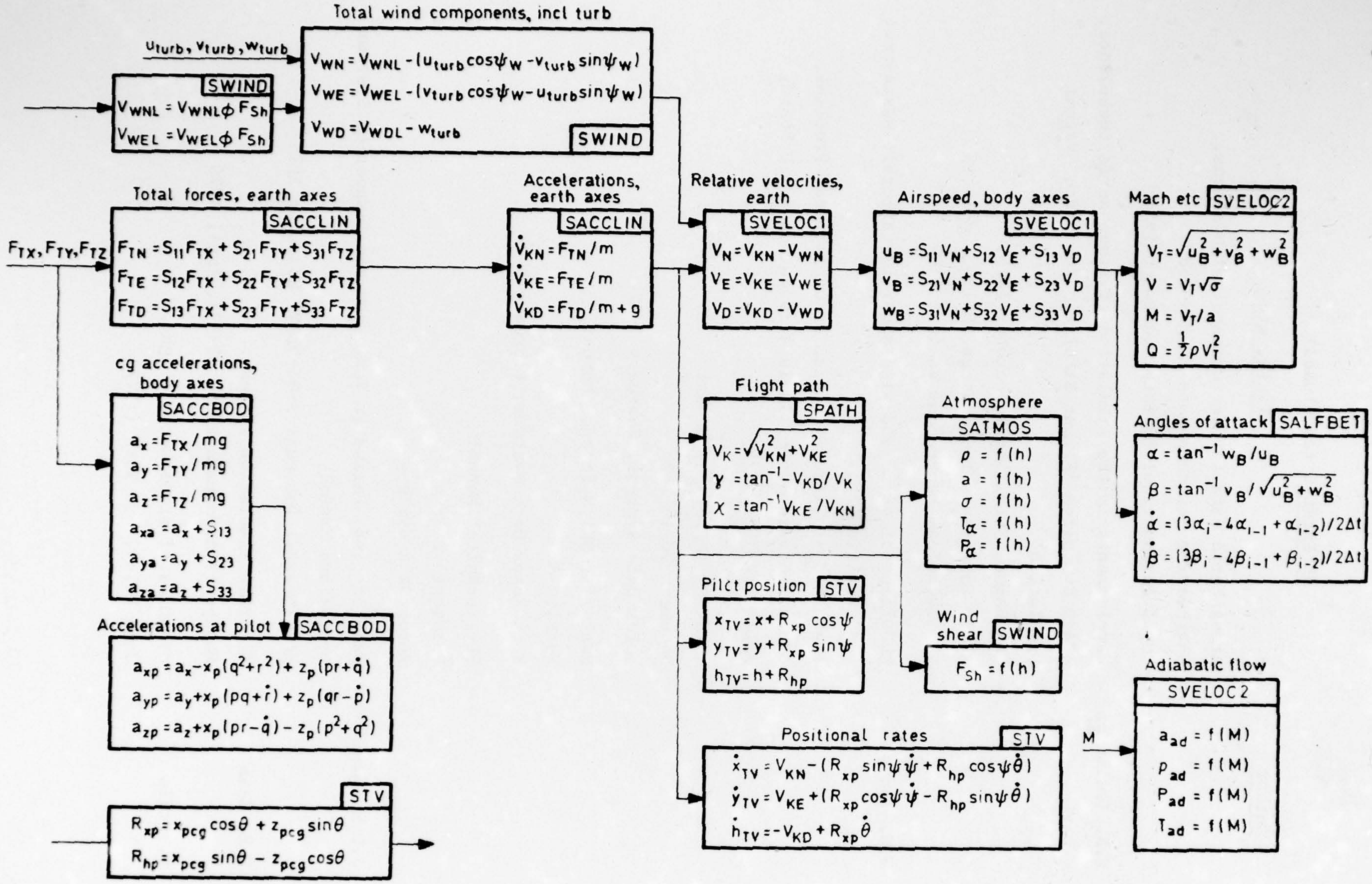


Fig 1 Block diagram of translational equations

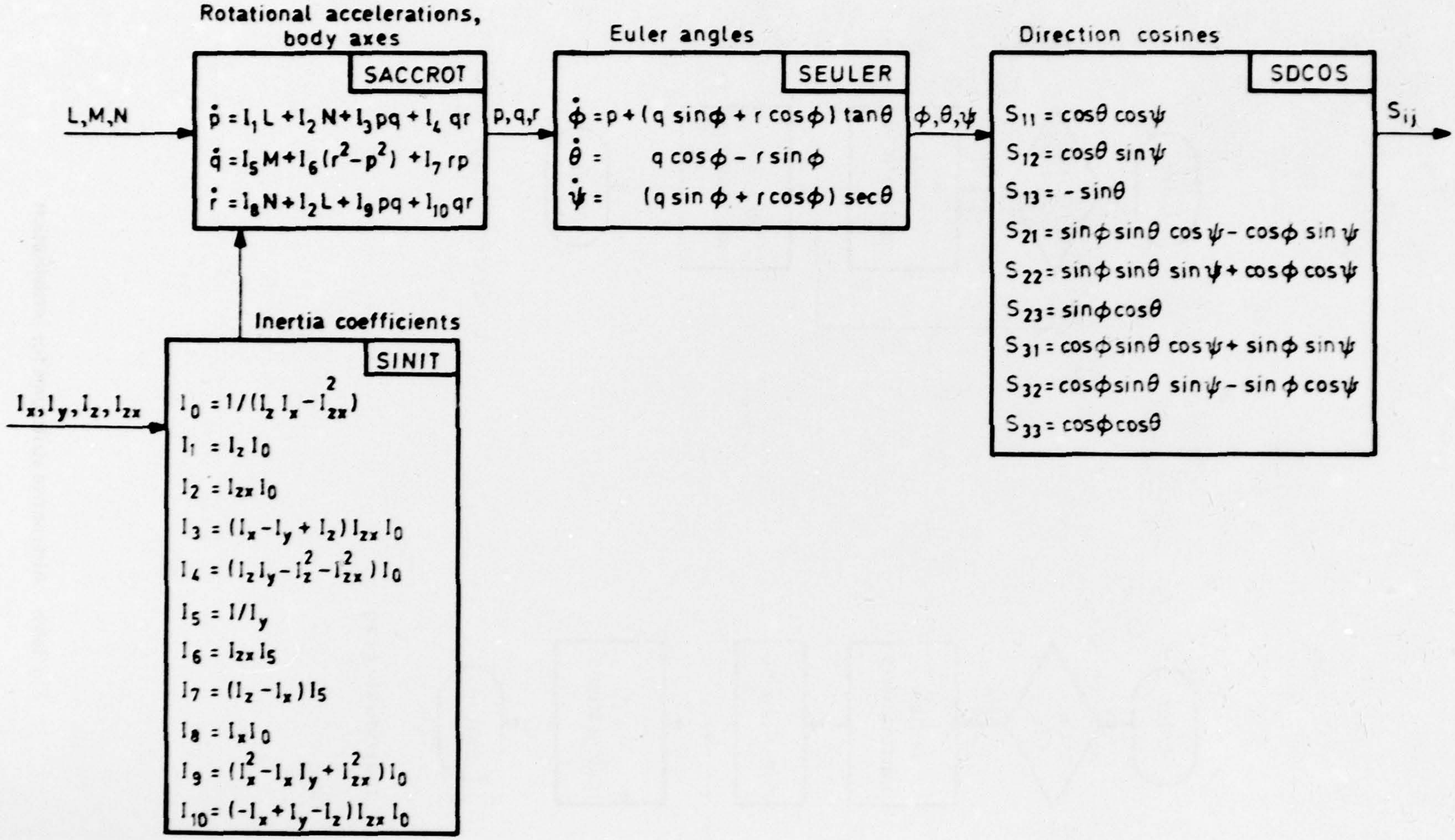


Fig 2 Block diagram of rotational equations

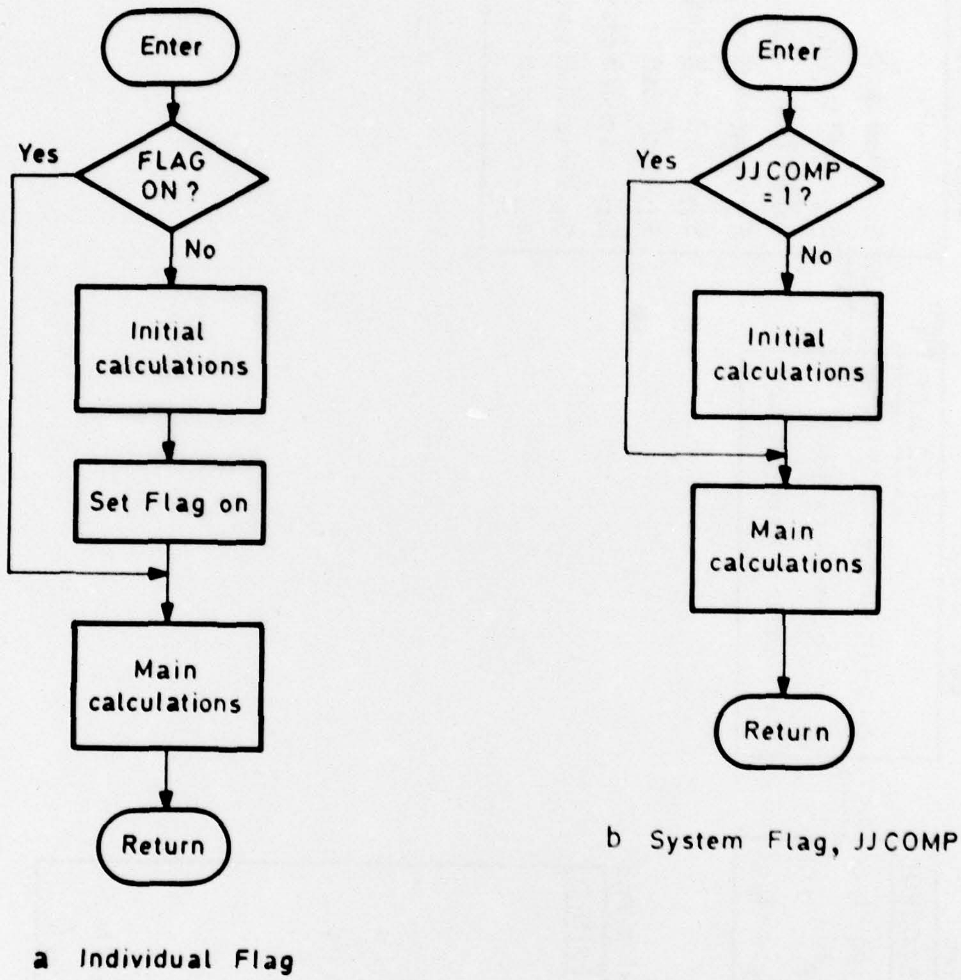


Fig 3a&b Alternative structures for initialisation

TR 79008

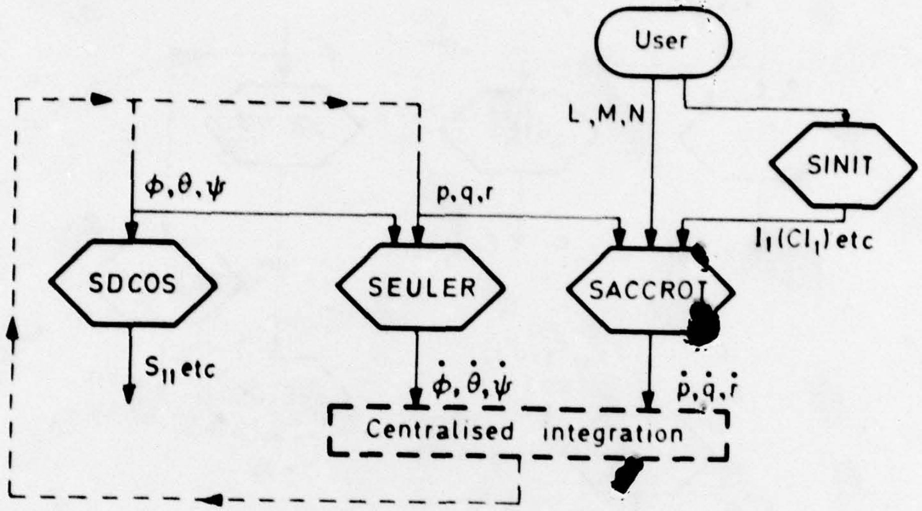


Fig 4 Information flow – rotational motion

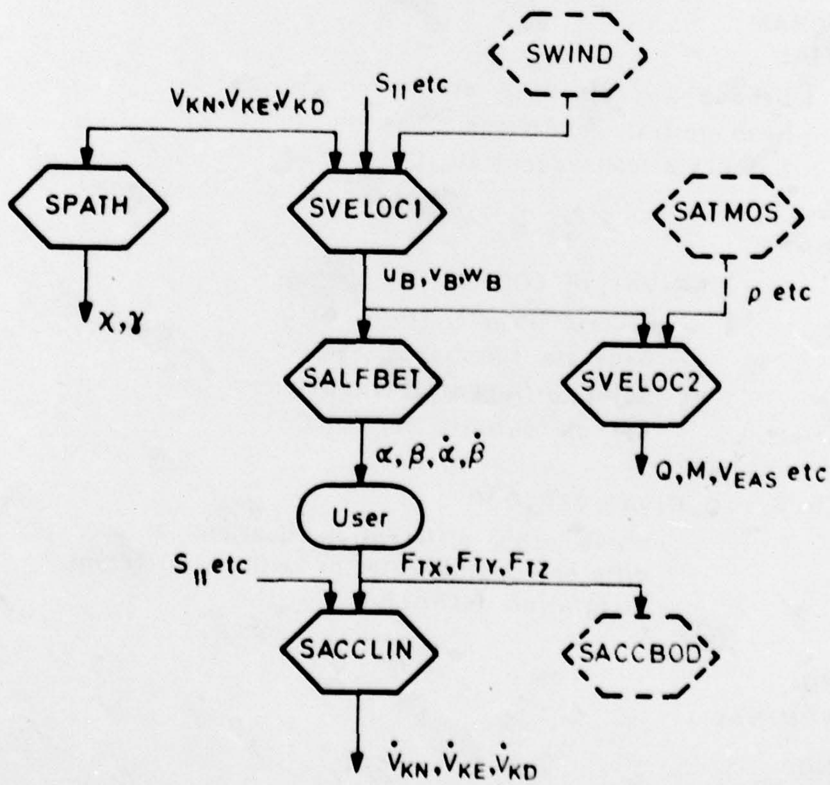


Fig 5 Information flow – translational motion

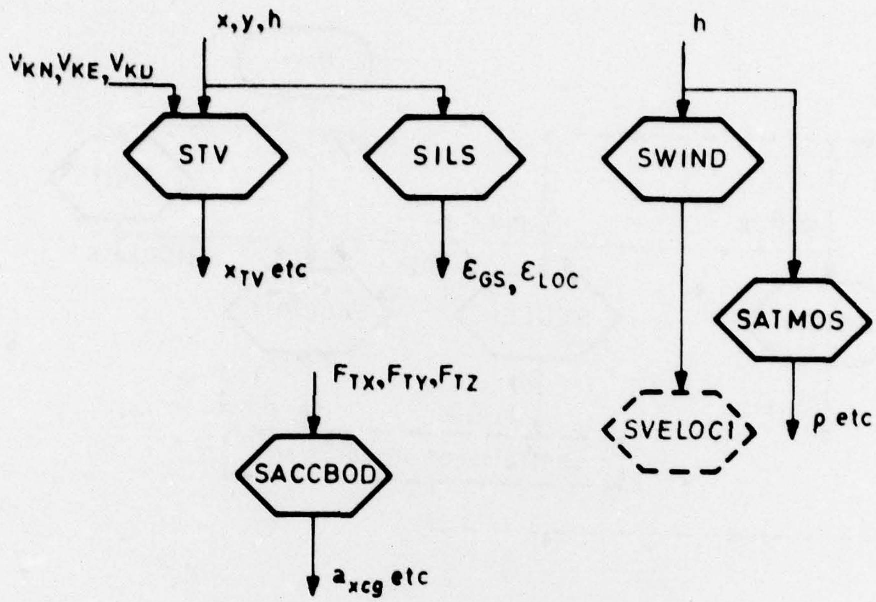
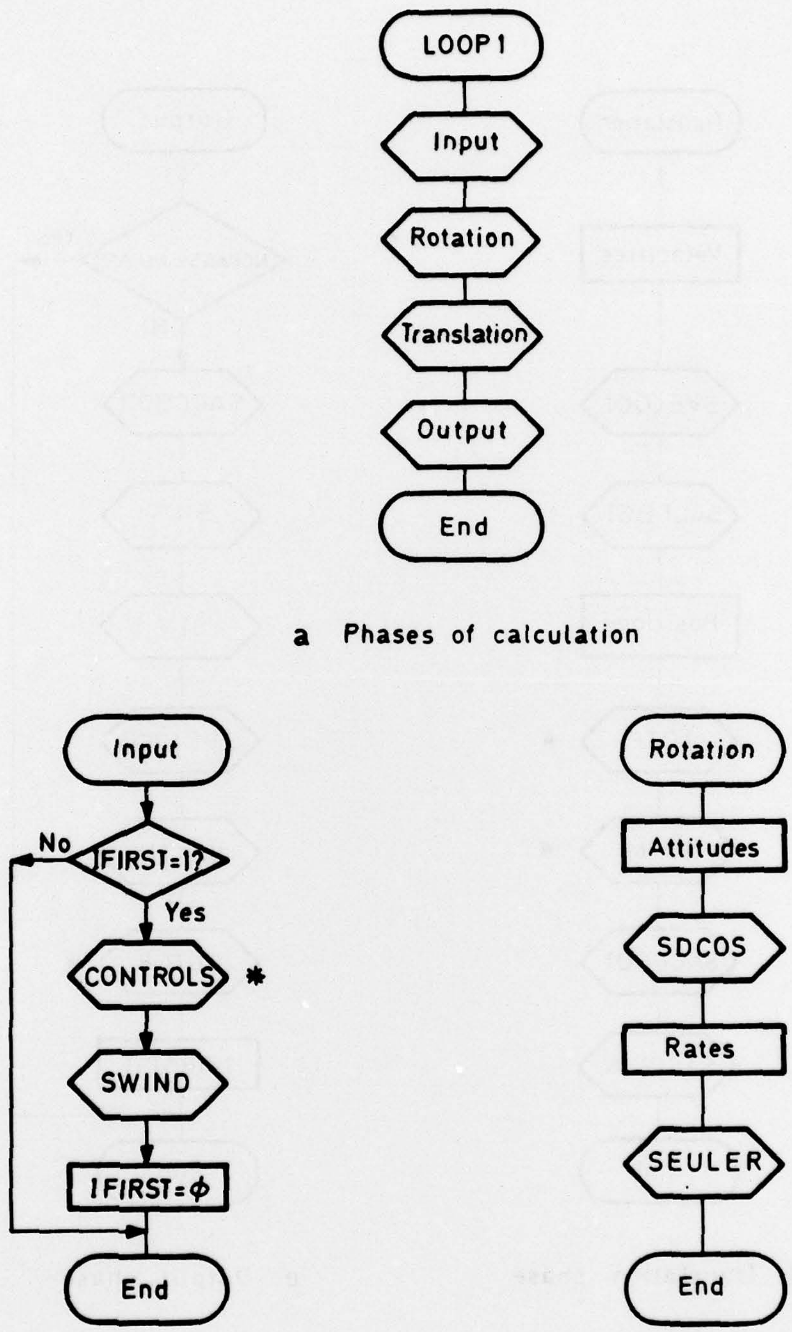


Fig 6 Information flow – utilities

```
PROGRAM
INITIAL
      Set up aircraft data etc
      Read initial conditions
      Calculate consequential ICs
END
DYNAMIC
          DERIVATIVE LOOP 1
               Receive inputs from ADC
               Generate functions
               Solve differential equations
               Set up outputs for DAC
          END
          DERIVATIVE LOOP 2
               Solve other differential equations at a
               different frame rate or with a different
               integration technique
          END
END
TERMINAL
END
END
```

Fig 7 SL1 model program structure



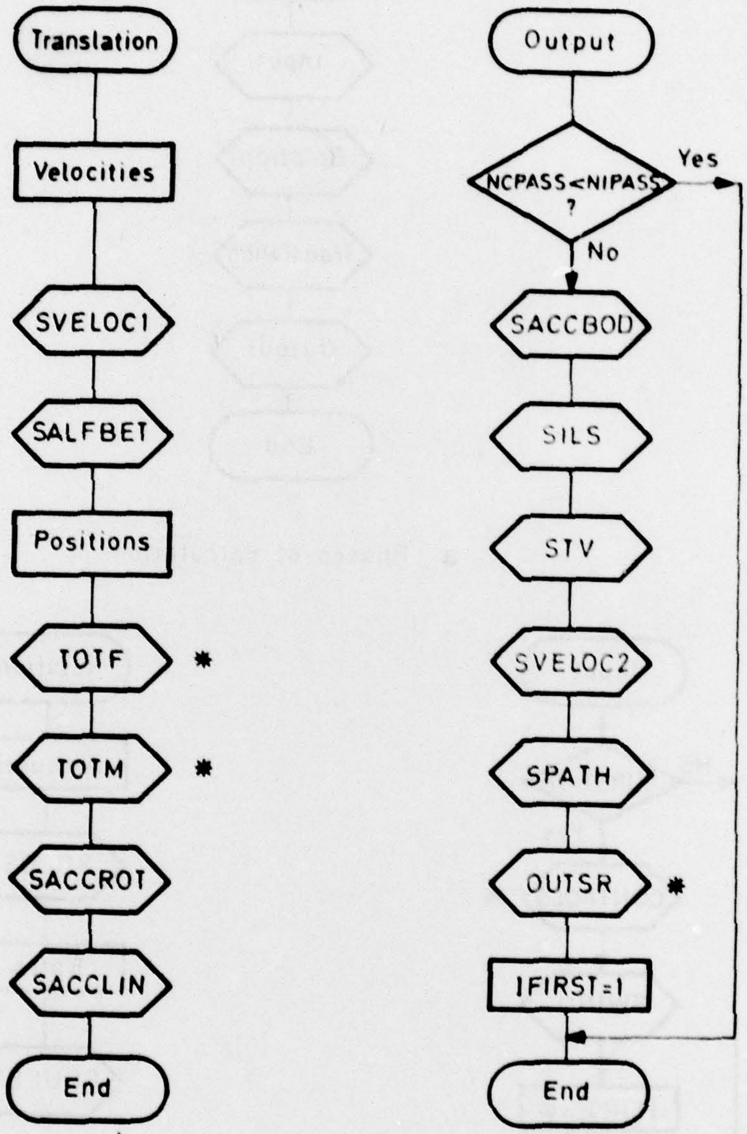
a Phases of calculation

b Input phase

c Rotation phase

* Indicates routine to be supplied by User

Fig 8a-c Execution sequence, single loop



d Translation phase

e Output phase

* Indicates routine to be supplied by User

Fig 8d&e Execution sequence, single loop

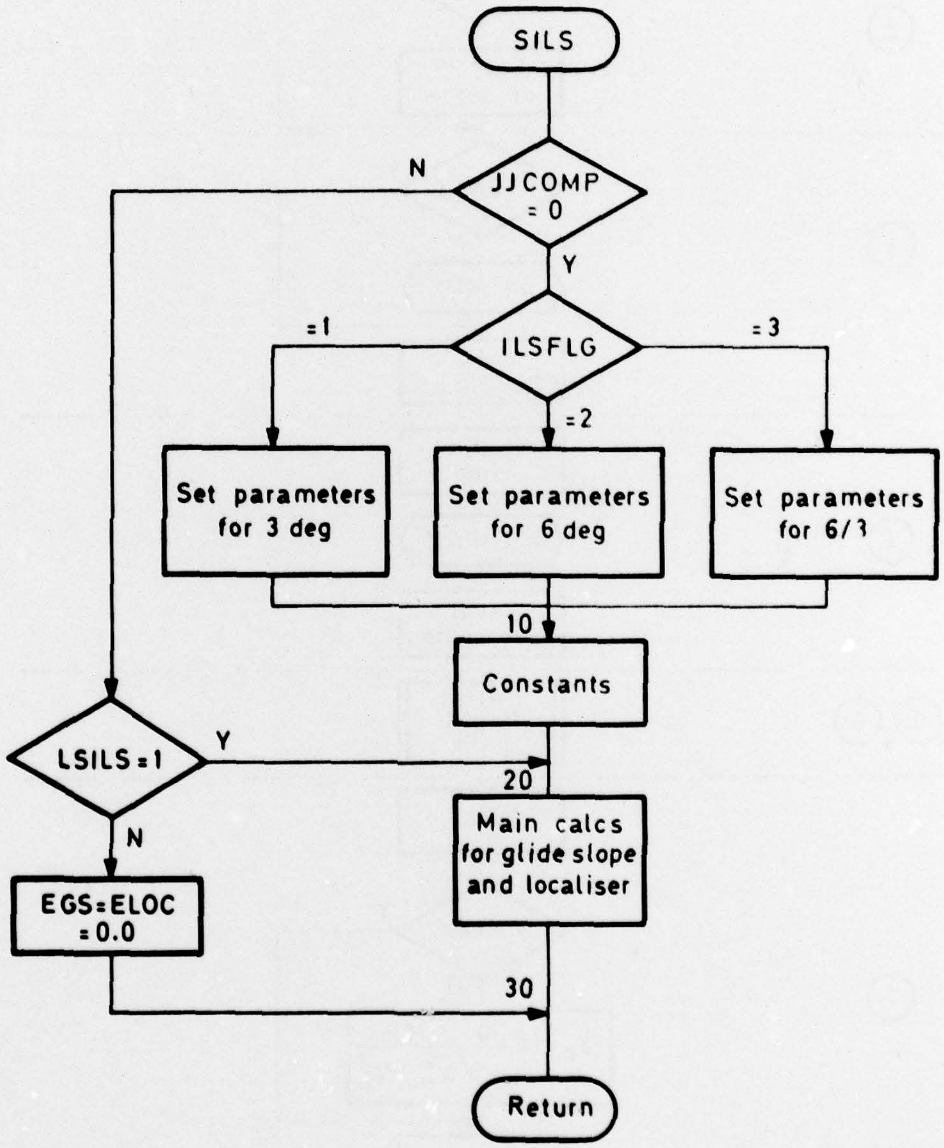


Fig A1 Flow diagram for SILS subroutine

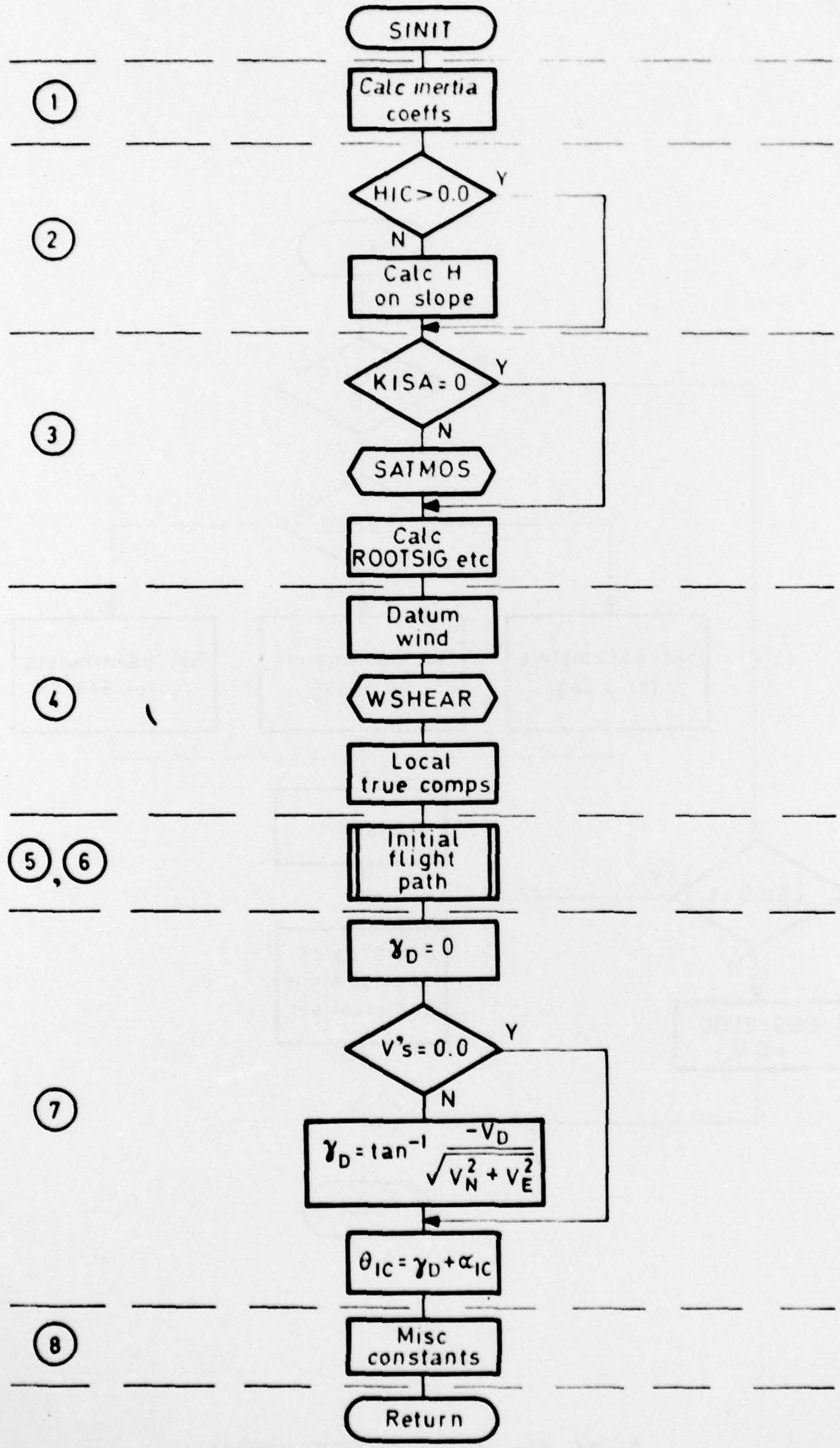


Fig A2 Flow diagram for SINIT subroutine

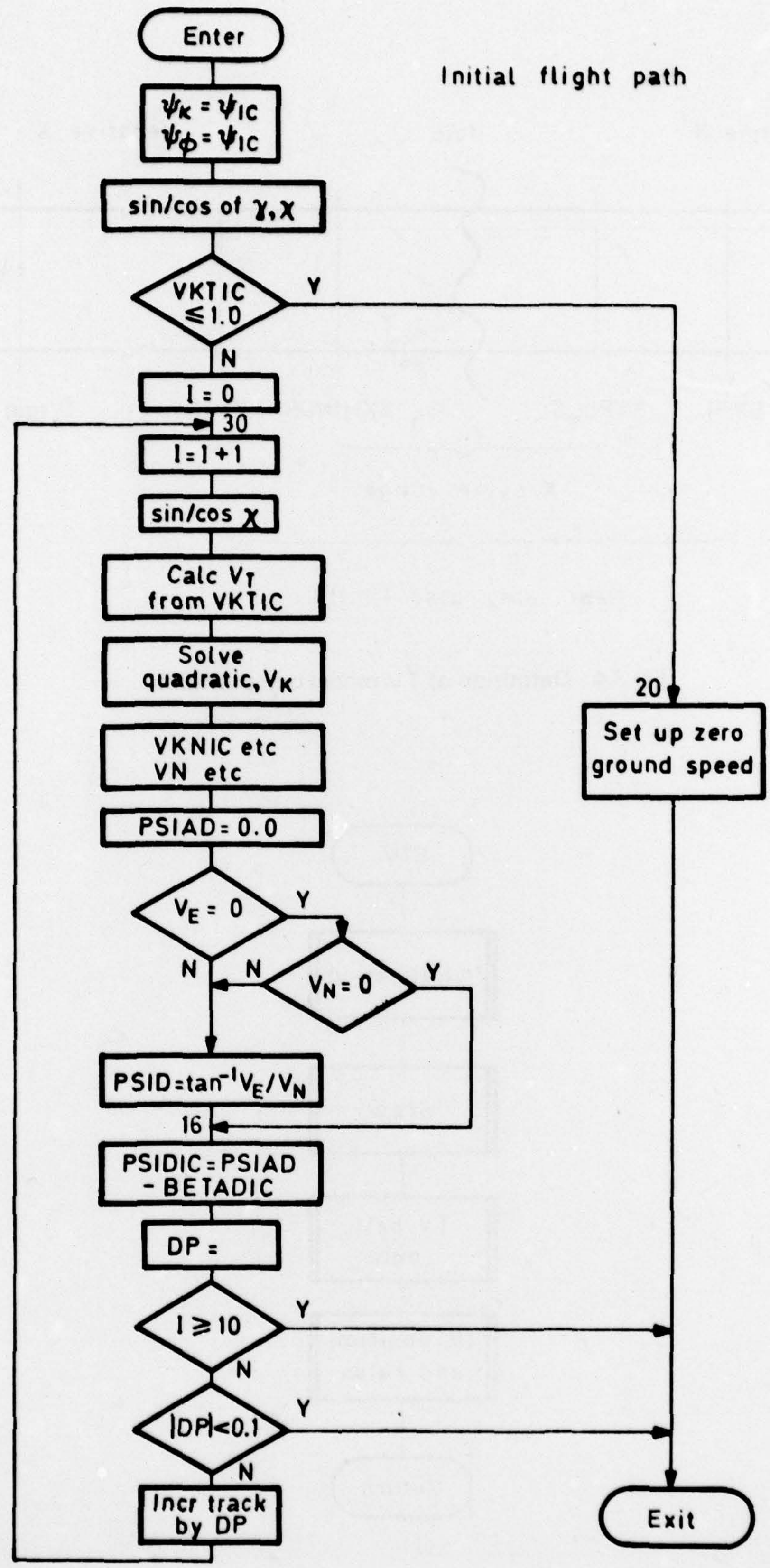


Fig A3 Flow diagram for 'initial flight path' phase of SINIT

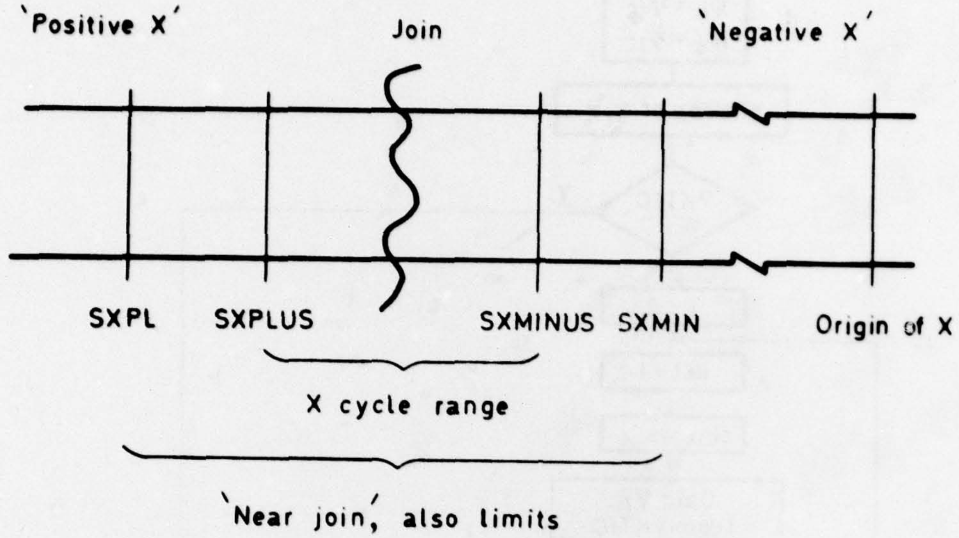


Fig A4 Definition of TV model belt positions

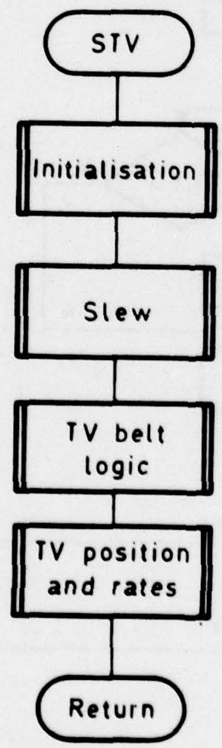


Fig A5 Overall flow diagram of STV subroutine

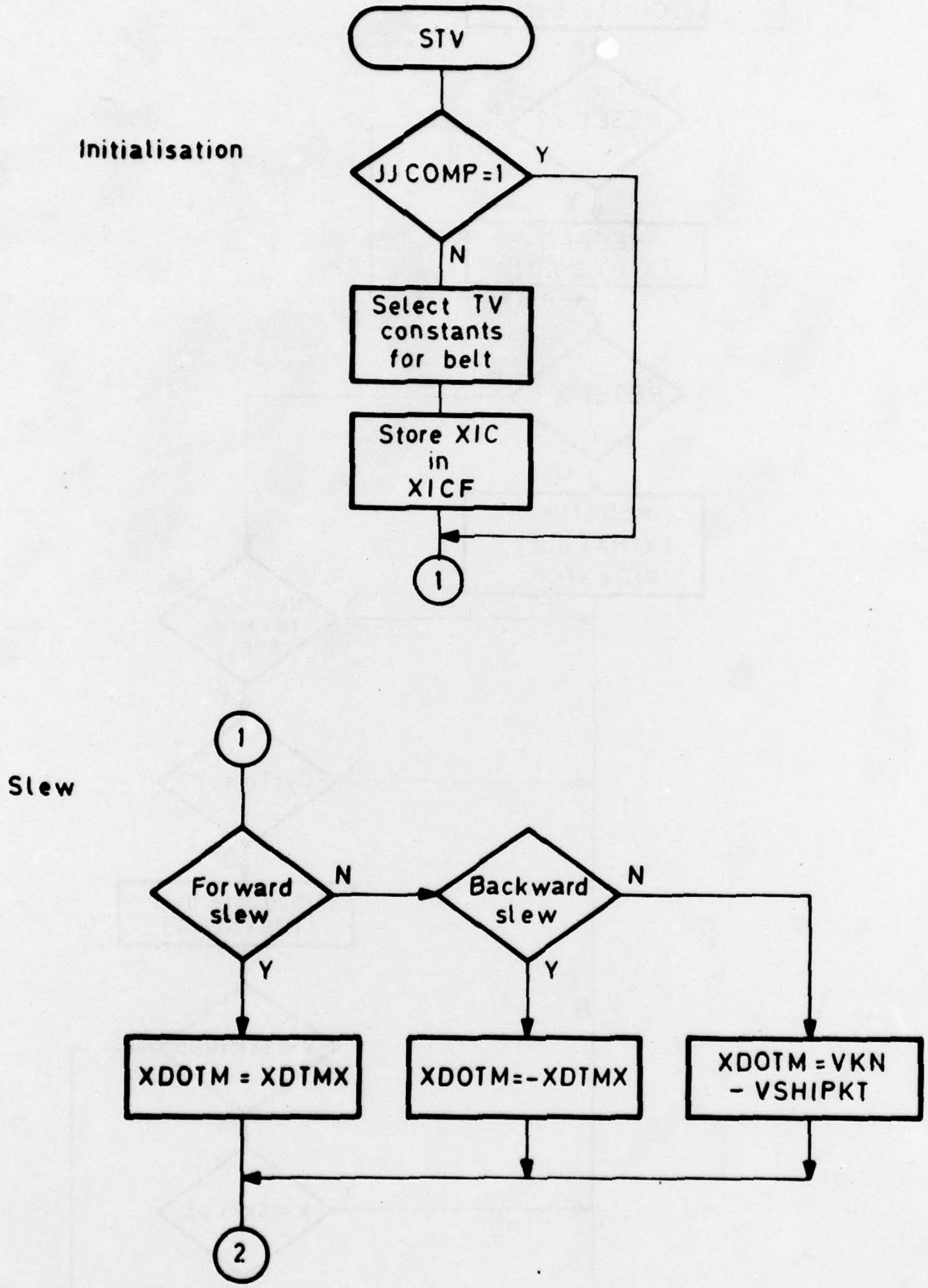


Fig A6 Flow diagrams of 'initialisation' and 'slew' phase of STV subroutine

Fig A7

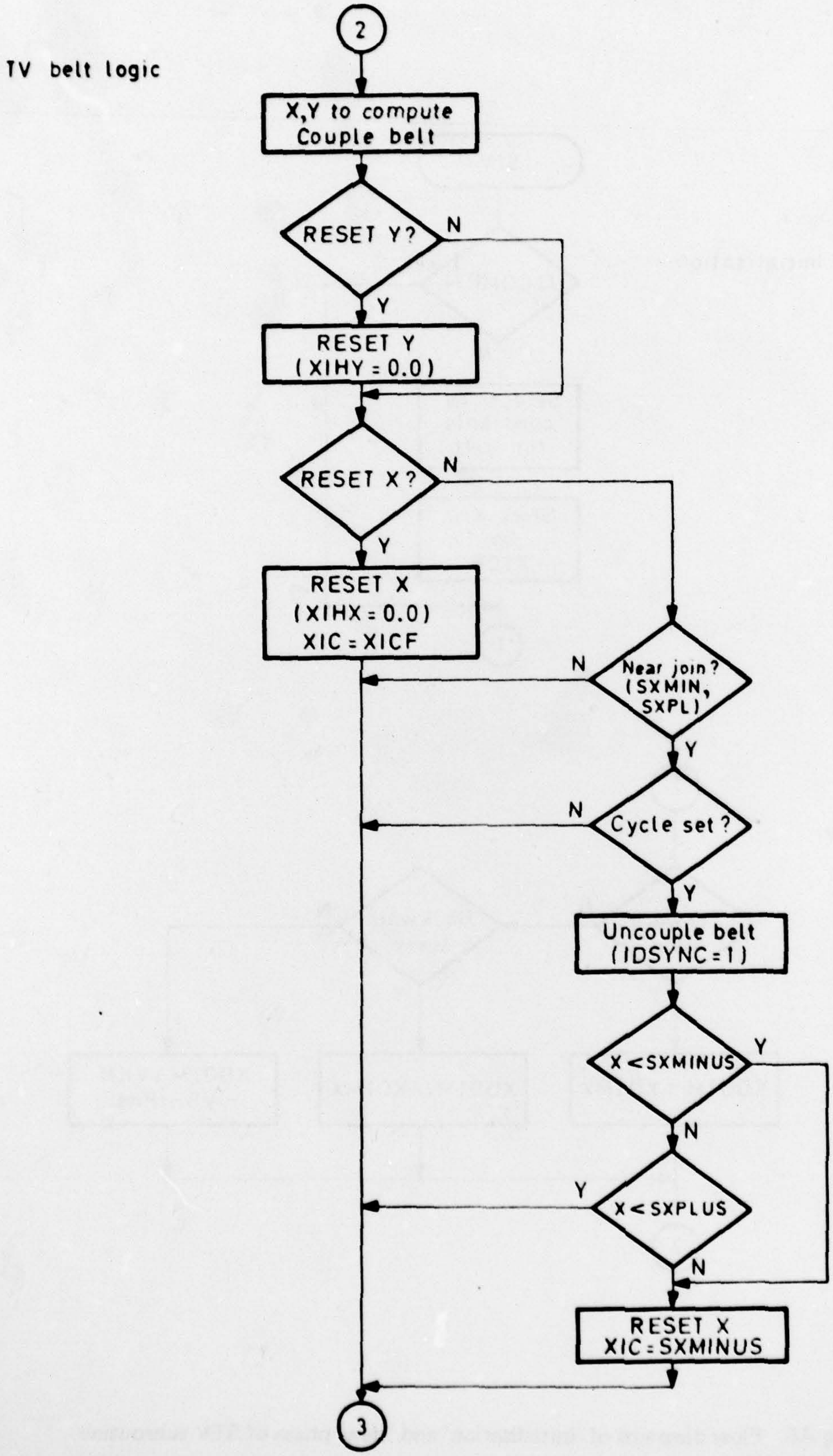


Fig A7 Flow diagram of TV belt logic in STV subroutine

TR 79008

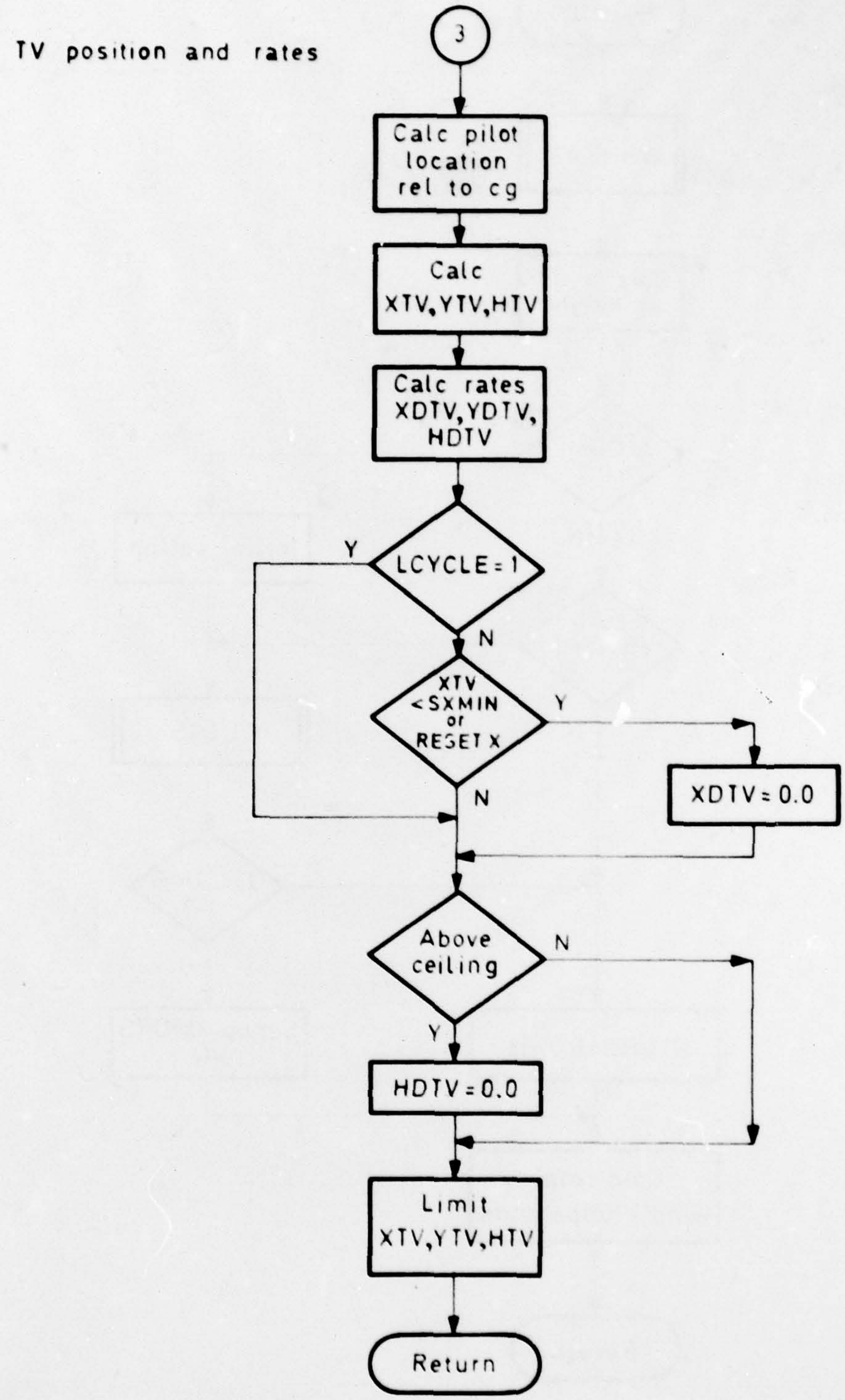


Fig A8 Flow diagram of TV position and rate control, STV subroutine

Fig A9

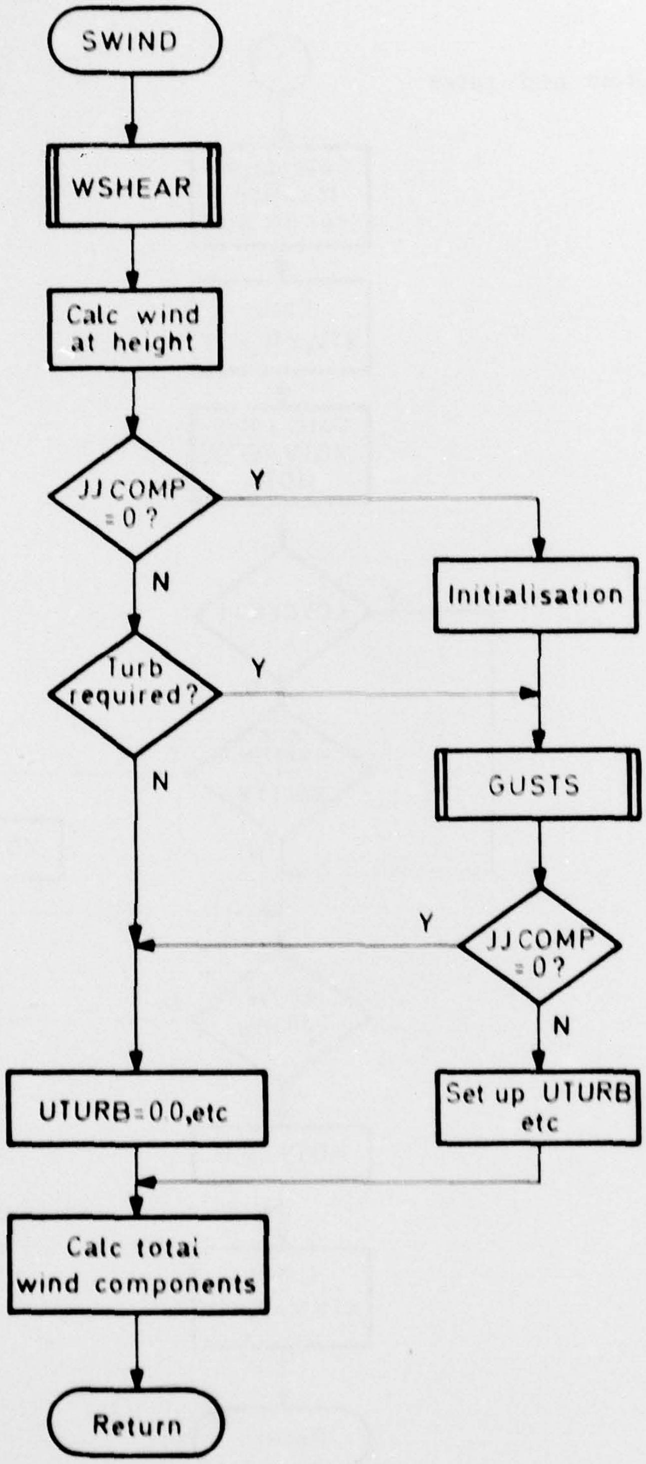


Fig A9 Flow chart of SWIND subroutine

TR 79008

REPORT DOCUMENTATION PAGE

Overall security classification of this page

UNCLASSIFIED

As far as possible this page should contain only unclassified information. If it is necessary to enter classified information, the box above must be marked to indicate the classification, e.g. Restricted, Confidential or Secret.

1. DRIC Reference (to be added by DRIC)	2. Originator's Reference	3. Agency Reference	4. Report Security Classification/Marking
	RAE TR 79008	N/A	UNCLASSIFIED UNLIMITED

5. DRIC Code for Originator	6. Originator (Corporate Author) Name and Location
7673000W	Royal Aircraft Establishment, Farnborough, Hants, UK
5a. Sponsoring Agency's Code	**6a. Sponsoring Agency (Contract Authority) Name and Location**
N/A	N/A

7. Title SESAME - A system of equations for the simulation of aircraft in a modular environment

7a. (For Translations) Title in Foreign Language

7b. (For Conference Papers) Title, Place and Date of Conference

8. Author 1. Surname, Initials	9a. Author 2	9b. Authors 3, 4	10. Date	Pages	Refs.
Tomlinson, B.N.			January 1979	149	19

11. Contract Number	12. Period	13. Project	14. Other Reference Nos.
N/A	N/A		FS 91

15. Distribution statement

(a) Controlled by – ~~Head of Flight Systems Department, RAE~~

(b) Special limitations (if any) –

16. Descriptors (Keywords) (Descriptors marked * are selected from TEST)

Simulation. Mathematical modelling. Software. Computer programs

17. Abstract

A system of equations has been developed for the simulation of an aircraft's motion in real time using a digital computer. Those parts of the mathematical model common to all aircraft have been created as a set of Fortran subroutines, leaving the user to create only a small group of routines specifically to describe his aircraft. The equations employed are defined and the computer implementation described in detail. The Report can be used as a handbook and "user guide" but as the routines described are not specific to real-time simulation they could be used as a basis for a general mathematical model of an aircraft for use on any computer which supports Fortran.

FS910/1